'68' MICRO JOURNAL

$2.95 USA

Australia	A $ 4.75	New Zealand	NZ $ 6.50
Singapore	S $ 9.45	Hong Kong	H $23.50
Malaysia	M $ 9.45	Sweden	30:-SEK

VOLUME IV ISSUE XII • Devoted to the 68XX User • December 1982

"Small Computers Doing Big Things"

YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or **8209** — nine-inch diagnonal screen

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

Pascal for 6809

Pascal for the 6809 is a true native code compiler. Unlike the usual P-code Pascals which run in an interpretive manner, ours produces efficient assembly language mnemonics which can be assembled and run directly. This compiler is available for both 6809 FLEX™ and UniFLEX™. Many features not found in other Pascal systems were implemented while avoiding those features completely non-standard. Features of the Pascal system include:

- Supports most of Jensen and Wirth specification
- Produces fast and efficient 6809, native code
- FLEX run-time package may be trimmed
- Double precision real numbers (16.8 digits)
- Implements scalar, subrange and structured data types
- Standard I/0 using file buffer pointers
- Dynamic storage allocation
- Ability to call other Pascal programs
- FLEX version may call assembly language programs
- Buffered or single character terminal input
- Standard math functions: SIN, COS, ARCTAN, EXP, LN, SQR, SQRT
- Random number generator function
- Many usable, sample programs included
- UniFLEX version supports:
  - Random file positioning
  - Ability to call various UniFLEX system routines
  - Ability to execute UniFLEX utility commands

Pascal on diskette for 5" and 8" 6809 FLEX is available for $200.00 The 5" version requires two disk drives. The UniFLEX version is $300.00 and includes one year of maintenance. All orders should include 3 percent for postage and handling (10 percent on foreign orders).

™FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.

technical systems consultants, inc.
111 Providence Rd., Chapel Hill, N.C. 27514
(919) 493-1451

'68'

MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-DMAF2-CDS1-CT82-Sprint 3
Southwest Technical Products
219 W. Rhap ody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Mainframe-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

CONTENTS

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
615 842-4600

68' Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage Paid ISSN 0194-5025 at Hixson, Tenn. and additional entries. Postmaster: send Form 3579 to 68' Micro Journal, PO Box 849, Hixson, Tennessee.

SUBSCRIPTION RATES
USA
1-Year $24.50 2-Years $42.50 3-Years $64.50
FOREIGN
See Page 52

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors **full name, address, date and telephone number.** It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands ONLY should be used (due to our proportional processor): .sp space, .pp paragraph, .fi fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continous text.

All articles submitted on diskettes should be in TSC FLEX™ format, either FLEX2 6800, or FLEX9 6809 any version.

If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). **Please use a dark ribbon!**

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word, 20 words or less $7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.

Intelligent Serial I/O Processor Board Now Available

The GIMIX Intelligent Three-port RS-232C Serial I/O Interface can significantly increase throughput of a multi-user system by reducing the number of interrupts between user terminals and the host CPU. The Intelligent I/O Board accomplishes this by buffering data transfers between system and users and preprocessing of the data.

Appropriate on-board software and operating system drivers are required. Software and drivers for OS-9 Level 2 will be available shortly from GIMIX.

- ✔ INDEPENDENT ON-BOARD **2MHZ 68B09** CPU
- ✔ UP TO **20K** OF ON-BOARD MEMORY (EPROM and RAM)
- ✔ BUFFERED DATA TRANSFER BETWEEN HOST AND ON-BOARD CPU USING A **Z8038** FIO I/O INTERFACE UNIT
- ✔ THREE **RS-232C** SERIAL I/O PORTS **(6551As)** WITH SOFTWARE SELECTABLE BAUD RATES, WORD LENGTH, STOP BITS, PARITY

Standard Version Including 4K RAM (Without Software) $438.11

— ***PARALLEL VERSION COMING SOON*** —

Uniflex For GIMIX Winchester Systems

TSC will be providing UniFLEX compatible with GIMIX Winchester systems. The NEW versions of UniFLEX for use with the Winchester systems will be delivered on 5" media as well as 8" media.

GIMIX 30 Pin Prototyping Board Now Available

- Double sided with plated thru holes and gridded power and ground lines.
- 8 rows of pads on .100 x .300 centers: up to 41 fourteen pin ICs.
- Accepts standard 6, 8, 14, 16, 20, 24, and 40 pin DIP devices.
- The entire top edge has pads for .100 x .100 header (ribbon) connectors.
- Pads for solder connections or .100 center headers on all 30 bus lines.
- Accepts 3 TO-220 regulators, 1 on the +8V & 1 ea. on the +/− 16V lines.
- Provisions for decoupling caps distributed throughout the array.
- Can be used with wire wrap, wiring pencil, solder wiring, etc.

With gold bus connectors and heat sinks (unassembled) $38.33

Now Available From GIMIX

(U.S. & Canada Only)

THE WINDRUSH EPROM PROGRAMMER

- ★ Probably the most versatile EPROM PROGRAMMER available. Interface & software for EXORcisor - II (fully addressable) and S-50 bus systems.
- ★ PROGRAMS AND VERIFIES 2508/2708, 2516/2716 (SINGLE AND TRI-VOLT TYPES) 2532, 2732, 2732A, 2564, 2764 and the 128K TMS2528 (16K x 8) - - - - - - - WITHOUT ADDITIONAL 'PERSONALITY' MODULES - - - - - - -.
- ★ PROGRAMMER extends out to your work area via 5' of twisted pair cable.
- ★ EXTENSIVE COMMANDS MENU... MOVE DATE, READ, PROGRAM, VERIFY EPROMS, EXAMINE/CHANGE BUFFER, FORMATTED DUMP OF BUFFER, FILL BUFFER.
- ★ Fully documented user's manual w/schematics & theory of operation. Professionally finished PCBs w/solder resist & component overlay.
- ★ SOFTWARE AVAILABLE FOR FLEX 2/9, SSB, OS-9 (LVL 1 NOW, LVL 2 LATER) and MDOS... All source files supplied. Specify disk size please!

NOTE: One version is supplied FREE. Extra versions: $25.00 each.

S-30 Interface/Programmer/Baseplate/Cable $375.00
EXORcisor Interface/Programmer/Baseplate/Cable $395.00

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

Build performance into your system

with OS-9** software tools

Unix*-based, multitasking, modular, and versatile: these key features are some of the reasons why more 6809 computer manufacturers have selected OS-9 as their standard operating system than any other. And OS-9 has been put to work by thousands of users in almost every conceivable computer application in business, science, industry, education, and government.

Your operating system should not be a barrier between you and your computer. OS-9 is very friendly and easy to use. Its modular structure makes it easy to customize, plus its comprehensive documentation shows you exactly how to interface it to just about any I/O device.

OS-9's advanced features unleash the performance potential of almost any 6809 computer — large or small. In many respects the OS-9/6809 combination is more powerful than many *minicomputers!*

There are two basic versions of OS-9. Both have the same basic features and capabilities. OS-9 Level One runs on small to medium sized systems having up to 64K memory. The Level Two version runs on medium to large size systems having memory management hardware and up to 1 megabyte of memory, and includes record and file locking for multiuser database applications.

Here are just a few reasons why you should insist on OS-9 for your microcomputer system.

- Over 40 utility commands
- Friendly "Shell" command interpreter
- Tree-structured multilevel file directories
- Full timesharing support with log-in and file security
- Fast, secure random and sequential access files
- Comprehensive English language error messages
- Compact real-time multitasking executive
- Hardware or software memory management
- Device independent interrupt-driven I/O
- Fully ROMable for small control systems
- Standard versions available from manufacturers of most popular 6809 computers

OS-9 PASCAL Language Compiler

- most complete and versatile PASCAL available for the 6809
- capable of generating P-code for interpretive execution while debugging OR
- highly optimized 6809 assembly language source code output for maximum speed
- "virtual memory" P-code interpreter lets you run large PASCAL programs

CIS COBOL * Compiler**

- ideal for most demanding business applications
- features ISAM, Debug, ACCEPT/DISPLAY and Interprogram Communications modules
- retains full compatibility with CP/M software
- meets ANSI 1974 Level One COBOL standard and is GSA certified
- Also available-FORMS 2 automatic program generator for easy interative design of screen oriented applications.

BASIC09 Structured Basic Interactive Compiler**

- fastest and most comprehensive full Basic language available for the 6809
- combines standard Basic with the best features of PASCAL
- features compiler speed interpreter friendliness and superlative debugging facilities
- option available: Run B...a ROMable run-time system for compiled Basic 09

C Language Compiler

- complete implementation of the UNIX version 7 C language
- includes INT, CHAR, SIGNED, UNSIGNED, FLOAT AND LONG data types, structures, unions, standard C library and a full preprocessor with macro definitions
- generates fully reentrant 6809 assembly language source code output

For more information contact your computer supplier, or

MICROWARE

Microware Systems Corporation
5835 Grand Avenue, Des Moines, Iowa 50312 515-279-8844 • Telex 910-520-2535

*Unix is a trademark of Bell Laboratories. ***CIS Cobol is a trademark of Micro Focus, Inc. OS-9** and Basic09 are trademarks of Microware and Motorola, Inc.

Software Catalog

MW OS-9 ASSEMBLER $125.00

A 6809 assembler specially designed for the OS-9 environment using Motorola standard instruction mnemonics.
For OS-9.

MW INTERACTIVE DEBUGGER $90.00

A useful tool for testing and debugging machine language programs or testing hardware.
For OS-9

TSC 6809 CROSS ASSEMBLER $100.00

This assembler runs on 6800 FLEX system with all the features of the 6809 assembler.
For FLEX.

TSC 68000 CROSS ASSEMBLER on 6809 $250.00

An assembler which runs on a 6809 FLEX system and supports full 32 bit math, macros, and conditional assembly.
For FLEX.

TSC DEBUG PACKAGE $75.00

This package is a complete assembler language program debugging tool capable of simulating the functions of the MPU.
For FLEX.

TSC RELOCATING ASSEMBLER/LINKING LOADER:
Just released for FLEX! $150.00

BUSINESS APPLICATIONS

INFOMAG Data Base Management 295.00

INFOrmation MAnagement system designed for any specific application for use by the non-computer expert. Capable of handling almost any collection of data.
Requires TSC XBASIC and FLEX.

RMS RECORD MANAGEMENT SYSTEM $200.00

Complete DMS in machine language. Uses include accounting, customer or personnel records, and situations which require data entry, data retrieval and update.

BILLPAYER w/source $189.95

A series of programs that were designed to fill the need of a household financial system or of a very small business.
In TSC XBASIC for FLEX.

MAILING LIST w/source $99.00

Enables the user to define and maintain mailing list oriented data bases.
Written for 6809 FLEX and TSC XBASIC.

DYNACALC Object only $200.00

Faster and more powerful than VisiCalc (tm)! An "electronic" spreadsheet developed to enhance your computer's productivity with a powerful software tool for planning, manipulating data, and probing alternatives.
Written for 6809 FLEX and UniFLEX [$395.00]

TABULA RASA FLEX $100.00
UniFLEX $110.00

Provides for the generating and maintenance of tabular computation schemes used for analysis of business sales and economic scenarios.
In TSC XBASIC for 6809 FLEX

INVENTORY W/MATERIAL REQUIREMENTS FLEX $100.00
UniFLEX $150.00

These programs enable the user to define and maintain inventories and include hierarchial materials requirements planning.
Written in 6809 for FLEX.

OSBORNE A/R FLEX $295.00

An open invoice system. Will print ageing reports, statements etc.
For FLEX and TSC XBASIC.

OSBORNE A/P FLEX $295.00

An invoice oriented system. Will keep track of your vendors and even print checks for you.
For FLEX and TSC XBASIC.

OSBORNE G/L FLEX $295.00

Uses double-entry posting to reduce off-balance situations. Can post to your accounts from A/P, A/R and the Cash Journal.
For FLEX and TSC XBASIC.

UTILITIES

AUTOTASK w/source $129.95

Consists of a set of memory resident commands that are aids in the design of user oriented applications programs.
Written in assembler for 6809 FLEX

TOOLKIT #1 Object only $49.95
w/source 9.95

Add EDIT to TSC BASIC's, along with DECOMPIL and cross reference.
Written in assembler for 6809 FLEX

TOOLKIT #2 Object only $49.95
w/source $69.95

A package of utilities and programs developed to extend the capabilities of the FLEX operating system.
Written in assembler for 6809 FLEX

EXTENDED UTILITIES: Object only $49.95
w/source $69.95

A set of 12 utilities that add the final touch to your utilities for FLEX.
Written in assembler for 6809 FLEX

PLOT: w/source $49.95

Designed to give you neatly formatted plot with the best resolution possible. Will plot histograms, bargraphs, XY plot and others.
Written in TSC XBASIC for 6809 FLEX.

SOME COMMON BASIC PROBLEMS: w/source $69.95

SCBP will allow you to figure income tax, tax depreciation, amortization schedules and more. From the Osborne book by the same name.
Written in TSC XBASIC for FLEX

SOME PRACTICAL BASIC PROGRAMS: w/source $69.95

SPBP like the above is from the Osborne book. The work of conversion is done for you.
Written in TSC XBASIC for FLEX.

PASSWORD Object only $69.95
w/source $89.95

Will enable you to create a system disk that cannot be booted without knowing the built in password.
Written in assembler for 6809 FLEX

UNIFLEX SIMULATOR FLEX $100.00
UniFLEX $110.00

Debug Uniflex assembler programs using TSC Debug and other facilities of FLEX.
For 6809 FLEX.

OS-9 SIMULATOR for FLEX: FLEX $100.00

Debug OS-9 assembler programs using TSC Debug and other facilities of FLEX. Requires OSM.
For 6809 FLEX.

HELP Object only $29.95
w/source $49.95

A date retrieve utility designed to save you time digging through manuals looking for info about computer language commands and statements.
Written in assembler for 6800 or 6809 FLEX

JOB CONTROL PROGRAM Object only $49.95
w/source $89.95

Allows you to enhance every other program you own.
Written for 6800 or 6809 FLEX

TSC BASIC: PRECOMPILER FOR 6800-6809 $50.00

Allows the user to write BASIC programs in a non-standard BASIC source format.
For FLEX.

TSC FLEX UTILITIES $75.00

A package of additional FLEX utility commands which includes memory dump, prompting delete to name a couple.

TSC FLEX DIAGNOSTIC $75.00

These utilities are designed for FLEX. Included in the memory diagnostics portion are zeroes and ones test, random pattern test and more. Disk examine, modify and test are also included.

TSC SORT/MERGE PACKAGE $75.00

A full-disk sort/merge which allows the contents of any size file to be sorted, including random files.
For FLEX.

READTAPE w/source $54.95
Requires a PIA

Will read TRS-80 level II BASIC tapes and convert programs to TSC BASIC.
Written for 6809 FLEX.

FULL SCREEN FORMS DISPLAY FLEX $50.00
UniFLEX $75.00

This package substantially extends the screen input/output capabilities by providing a table driven method of describing and using full-screen displays.
Written for 6809 FLEX.

GAMES FOR FLEX

ESTHER Object only $39.95
w/source $59.95

An enhanced game of Eliza in fast machine language. Artificial intelligence in pure 68XX code.
Written in assembler for 6800 or 6809 FLEX

NEW for OS-9!!

DISK/EDIT: $79.95

Examine and modify OS-9 DISKS with this screen oriented editor.
Written for OS-9.

For FREE Catalog
send legal size SASE with 3.40 stamps.

ORDERING INFORMATION

SPECIFY 5 OR 8 INCH DISKS - 6800 or 6809

$2.50 for USA SHIPPING and HANDELING

20% or $20.00 for FOREIGN AIR - EXCESS REFUNDED

VISA, M/C & DINERS CLUB cards accepted.

FHL FRANK HOGG LABORATORY

THE REGENCY TOWER
SUITE 215, 770 JAMES STREET
SYRACUSE, NY 13203 (315) 474-7856
TELEX-646740

Software Catalog

OPERATING SYSTEMS

FHL Color FLEX — Object only $99.00

FLEX for the Color Computer! All the power and capabilities of the FLEX operating system for the RS Color Computer.
For 64K Color Computer

LANGUAGES

O/BASIC — with FHL Color FLEX $30.00; later $40.00

Radio Shack 019K BASIC for FHL FLEX. O/BASIC allows the use of the standard Disk Extended Color Basic under FHL Color FLEX. All disk input and output operations are done through FLEX and are completely compatible with the normal FLEX utilities.
For FHL Color FLEX only.

TSC BASIC for 6800-6809 — $75.00

Supports all of the standard BASIC statements and functions as well as many extended capabilities. For FLEX.

TSC EXTENDED BASIC FOR 6800-6809 — $100.00

Ideal for business or advanced scientific applications where extended math precision and formatting capabilities are essential. For FLEX

TSC PASCAL for 6809 — $200.00

A true native code compiler which produces assembly language mnemonics. For FLEX.

TSC FORTRAN '77: — $275.00

Just released for FLEX - Real-time Relocating assembler. $350 with relocating assembler and linking loader. For FLEX.

MW BASIC09 — $200.00

Acclaimed as the most powerful and friendly high-level language available for any microcomputer. For OS-9.

A/BASIC Compiler: — Object only $150.00

Generates pure, fast, efficient 6809 machine code from easy to write BASIC source programs.
Written for 6809 OS-9 or FLEX

X-FORTH — Object only $149.95

An enhanced and extended version of Fig-Forth that runs under FLEX.
For 6800 or 6809 FLEX.

CC-FORTH — $99.95

For the Color Computer 09S, a language that is 5 to 10 times faster than BASIC.

DYNASOFT PASCAL — Object only $69.95; w/source $99.95

A fast integer only P-Code compiler ideal for CBASIC applications. Powerful enough for the DynaStar word processor.
Written for 6809 OS-9 ($10 less for FLEX)

MW OS-9 PASCAL COMPILER — $400.00

A comprehensive implementation of PASCAL. It has ability to generate either highly optimized assembly language source code or P-code for interpretive execution while debugging.
For OS-9.

MW CIS COBOL COMPILER — $895.00

The "CIS" stands for compact, interactive, and standard. Making COBOL ideal for microcomputer business applications. For OS-9.

MW FORMS 2 FOR CIS COBOL — $200.00

A COBOL program generator which facilitates fast and convenient development of interactive screen oriented applications.

WORD PROCESSING

DYNASTAR — Object only $149.95

Screen Editor: A powerful menu-driven screen editor equally suited to the tasks of program preparation and document processing. Similar to WORDSTAR (tm).
Written for 6809 OS-9 and FLEX.

DYNAFORM — Object only $149.95; With DynaStar, both only $275.00

Text Formatter: Provides all the standard features such as pagination, headers and footers with page numbers, single space, double space, bold-face, double strike and more.
Written for 6809 OS-9 and FLEX.

DYNASPELL — Object only $199.00; w/source $299.00

The most versatile 6809 spelling checker available. Fast and easy to use.
Written in assembler for 6809 OS-9 or FLEX

STYLOGRAPH 2.0: — $295.00

SPECIAL NOW FOR FHL COLOR FLEX ONLY — $195.00

A very powerful, yet easy to use text processing system. This system is cursor-oriented with dynamic screen formatting.
For 6809 FLEX and OS-9.

STYLOGRAPH MAILMERGE — $125.00

For use with Stylograph Word Processing System, will fulfill three different needs.
For 6809 FLEX and OS-9.

STYLOGRAPH SPELLING CHECKER — $145.00

A fast and very easy to use tool that allows even an inexperienced user to eliminate spelling errors from text!
For 6809 FLEX and OS-9.

READTEST — Object only $54.95; w/source $74.95

A must for all writers and instructors. It reads your text and tells you how well it was written.
Written in assembler for 6800 or 6809 FLEX

TSC TEXT PROCESSING SYSTEM — $75.00

This system allows the use of over 50 commands for special text formatting applications.
Post processor for FLEX.

SOFTWARE DEVELOPMENT TOOLS

CRASMB — FLEX $139.95; OS-9 $200.00

CPM's (CPU Modules) — FLEX $25.00; OS-9 $35.00

Source or binary for the following: 6800, 6801, 6809, 6502, 1802, Z80, and Z8. OS-9 includes 6809 binary.
Will cross assemble source code into object code. (runs on 6809 systems.)
Written for 6809 FLEX and OS-9.

DSM — $125.00

OS-9/FLEX MACRO ASSEMBLER: A fast and versatile macro assembler with ability to define macros, with substantial parameters, conditional assembly directives and ability to change value of a label or symbol. Create OS-9 binary files in FLEX and vice versa!
Written for 6809 OS-9 or FLEX

SUPER SLEUTH — 6800/6809 w/source $99.00; Z80 w/source $99.00

A set of programs which will enable the user to examine and/or modify binary program files on disk or in memory.
Both written for FLEX, UniFLEX and OS-9.

6502 TRANSLATOR w/source — FLEX $75.00; UniFLEX $80.00; OS-9 $85.00

Enables the user to translate 6502 assembler code into 6809 assembler code.
Written for 6809 FLEX, UniFLEX and OS-9.

DEBUGGING SIMULATORS 6805 or 6502: w/source — FLEX $75.00; UniFLEX $80.00

Programs which enable user to simulate, examine and/or modify object 6805 and 6502 program files on 6800 and 6809 systems under FLEX.
Written for 6809 FLEX and UniFLEX.

PIC/PID 6800 Translator: — FLEX $90.00; UniFLEX $80.00; OS-9 $75.00

Translates 6800 assembler programs to 6809 mnemonics and converts 6809 programs to position independent code and data (PIC/PID)
Written in assembler for the 6809.

CROSS ASSEMBLER MACROS: — FLEX $50 ea - 3/$100; UniFLEX $60 ea - 3/$120

6800/1, 6502, 6805, 8080/5 and Z80: For use with the TSC assembler.
A macro text file.

DSM - OS-9/FLEX MACRO ASSEMBLER — $125.00

For FLEX or OS-9, Create FLEX or OS-9 binary files from either FLEX or OS-9. DSM is a MACRO assembler like C ASMB, it is similar to TSC's Assembler, but it has more powerful MACROS. DSM makes it easy to move FLEX programs to OS-9. In OS-9 it gives MACRO capability like TSC's assembler and is compatible with TSC source files. DSM was used by the author to move CRASMB to OS-9.
For OS-9 or FLEX.

EDitor — $50.00

ED has all the features of TSC's editor with the addition of screen type editing, MACRO capability, and a math package. Works with files larger than memory. It has many additional features.
For 6809 FLEX.

ASM - assembler — $90.00

ASM is compatible with TSC's assembler. But with more powerful MACROS and conditionals, than TSC's.
For 6809 FLEX.

MW MACRO TEXT EDITO — $125.00

Combines a minimum keystroke text editor with a macro-driven string processing language. A powerful tool for creation, conversion, or reformatting.
For OS-9.

BOMBED?

For over four (4) years now we have published every month, a 68XX magazine devoted to the Motorola 68XX series of microcomputer devices. Independent surveys have indicated that we, 68 Micro Journal, reach more Motorola 68XX users, than any other magazine. One good reason for this is that we have attempted to keep you informed. In all that time, and on all those thousands of pages, I have stated that we believe that the 68XX series of devices are the **finest in the world!** I have not waived from that conviction.

Please bear that in mind as you read the following.

A few months back we published a letter concerning some problems experienced by a 6809 user. It seemed that this letter was some two years old. It got mixed into current material by accident, during the clean up from a flooding of our offices. It appeared, at first, that the complaint was an isolated instance (defective 6809), and should not have been run. However, I soon began to receive communications from other users, indicating that they also had experienced problems with bad 6809 microprocessors. This and information mentioned in the first article by Dr. Graves on "Simple Winchester Interface, October 1982" triggered a small wave of input from other readers, indicating that they finally fixed problems that had befuddled them for, in some instances, years (see 68 Micro Journal, November 1982). We had even experienced problems with defective 6809 devices, however, at the time we did not realize that it could be the 6809. This was caused by the problem appearing after we had installed a 'winchester' disk system to the computer. We, as many others, attributed the problem to the disk.

In our case we lost a mailing list that contained over 48,000 entries. Parts of the data was backed off to some 8 inch floppies, which saved many days of reentry. But, still we spent weeks getting the file back to current data. Like NASA we had been 'floored' by an inexpensive part, a $15.00 6809 chip.

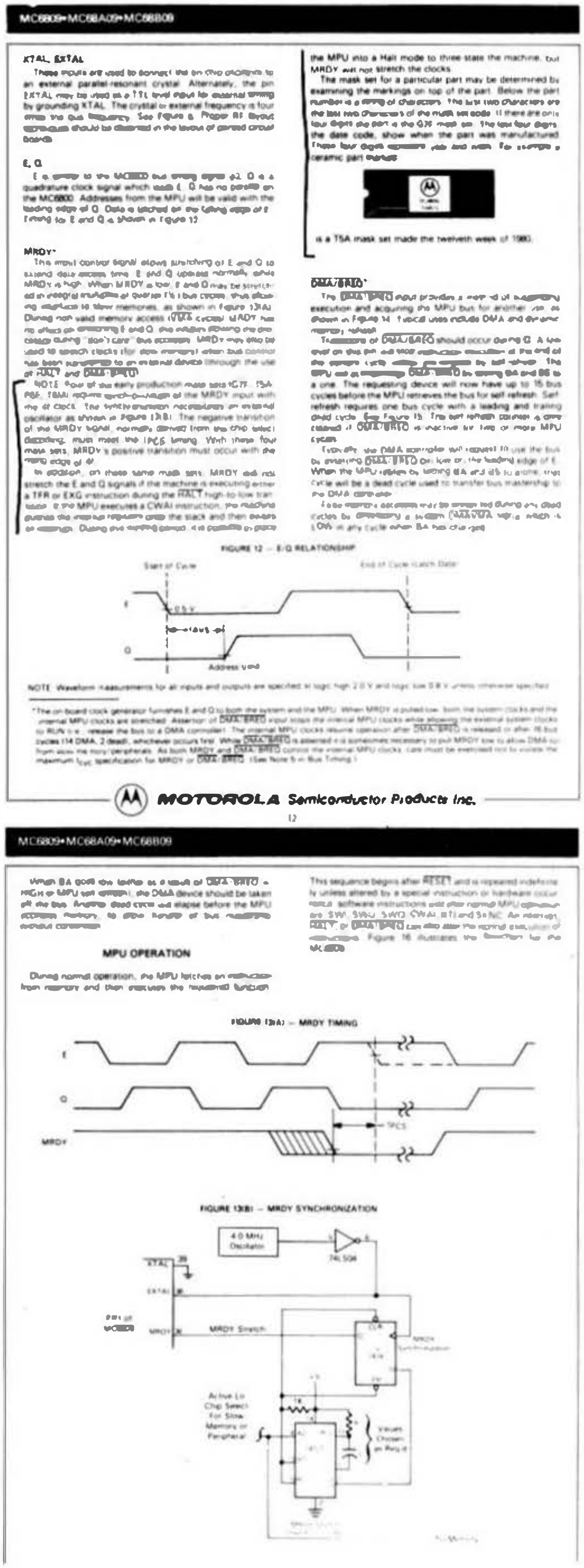
Since the first mention of this problem (October 1982 68 Micro Journal) I have received many reports that lead me to believe that there are a large number of you out there that are having similar problems. Also I felt that Motorola might offer some type of exchange for the defective parts, however, as you can see from the reply below, from Motorola, there is no indication that this will happen.

I can well imagine that to offer some sort of exchange or credit towards a 'good' 6809 could entail a not so small expense to Motorola. Whatever the reason I still feel that I have, as stated in our first issue over 4 years ago, an obligation to you, our readers, to let you know of problems that could affect the proper operation of your 68XX equipment. Also I want you to know that once I brought this problem to the attention of Motorola (see copy of letter below) the response from Motorola has been both prompt and open. Engineers at Motorola, that I have discussed this with, have in every instance been completely frank in their appraisal of this situation. So despite the fact that I am disappointed in the response from Motorola, I still believe that the **Motorola 68XX series of devices are the finest in the world!**

Again for your information the following 6809 mask numbers are to be avoided. They should be replaced with either the 'CW3' or 'GF7' series of 6809 microprocessors.

G7F T5A P6F T6M W8L (W8L may be o.k.)

For those who wish to modify their existing CPU with the recommended Motorola 'fix' the following 'official' data is reproduced from a Motorola publication.


MC6809•MC68A09•MC68B09

XTAL, EXTAL

E, Q

MRDY*

is a T5A mask set made the twelveth week of 1980.

FIGURE 12 — E/Q RELATIONSHIP

MOTOROLA Semiconductor Products Inc.

12

MC6809•MC68A09•MC68B09

MPU OPERATION

FIGURE 13(A) — MRDY TIMING

FIGURE 13(B) — MRDY SYNCHRONIZATION

NOTE: For those users who have 6809 systems that do not 'stretch' the clock, the need to replace the 6809 is less compelling. The problems seem to occur mainly when the 'E' or 'Q' clock signals are stretched.

One thing that bothers me in this entire affair is that some manufacturers of 6809 equipment have taken, from some of their customers, a bum rap, because of a lack of this information. Also since this has come up I have received information concerning other problems. These are being looked into, especially concerning the 6809 and 68000. If there is evidence that there is information, that you the user need to know, then we will publish it as soon as I can get verification.

Murray A. Goldman — Vice President, General Manager, Microprocessor Division

October 13, 1982

Mr. Don Williams, Sr.
Publisher
'68' Micro Journal
P. O. Box 849
5900 Cassandra Smith Road
Hixson, Tennessee 37343

Dear Mr. Williams:

Thank you for your letter, dated September 28, pertaining to the MC6809 mask variations and other issues. I will attempt to answer each of the major points of your letter.

The communication of specification addenda to an extensive customer base is a challenge not only to Motorola, but to the industry as a whole. In part, this is due to the large number of customers that purchase products "off the shelf" from distributors. Since this is a very common practice, vendors such as Motorola have little, or no visibility to a very significant portion of their customer base. This lack of visibility is exacerbated when a large international distribution network is considered. Because of this lack of visibility, I am pleased that '68' Micro Journal is intent to convey this important information to many customers whom we cannot reach. I do not believe that Motorola is guilty of hiding information from customers, rather, we are unsuccessful in getting the information to all of the customers. The MC6809 can serve as an excellent example.

The MC6809 was introduced in 1978. In late 1979, a routine customer applications inquiry led to the discovery of the MRDY deviation. The advanced data sheet was modified and reissued in early 1980. The revised document clearly and explicitly advised customers of the deviation. This information was also included in the data manual issued in 1981. Meanwhile, the part was modified to make the part consistent with the original design intent. The W8L mask set was released to production in February, 1980. These parts seemed to satisfy customers but were difficult to make. As a result, the present CW3 mask set was generated and released to production in July of 1980. Our factory in Scotland is also making a "clean" version using their GF7 mask set.

Considering the level of effort required and the mask fabrication and wafer processing time, the turnaround was fast by anybody's standards.
You mentioned that there were numerous "mask runs" of the 6809, all with this or other problems. Your information is technically accurate (Motorola's engineers are delightfully honest) but, out of context, conveys an inaccurate connotation. A brief review of the mask set evolution will elucidate my contention.

Mask Set	Reason for Change
G7F	Initial Mask
T5A	3 to 4 Inch Wafer
P6F	NMOS to HMOS
T6M	Yield Enhancement (MRDY Deviation Discovered)
W8L	Change due to MRDY
CW3 (U.S.); GF7 (Scotland)	Existing Mask Set

Notice that the MC6809 was in its 4th normal mask progression when the MRDY issue was discovered. Rapid mask variations are an important part of the learning curve that we have all come to expect and demand.

Motorola is not alone in experiencing difficulties such as MRDY in the MC6809. Experiences like the 6809 MRDY issue are not unusual in the industry. Motorola takes every action possible to assure conformance to the letter and intent of the specification. The initial functionality investigation is quite lengthy. Consider, for example, one aspect of this investigation, instruction sensitivity. There are 1464 different variations of instructions and addressing modes in the MC6809. Taking these instructions two at a time and then verifying that all 2,141,832 combinations function as expected can be quite time consuming. Unfortunately, this does not assure that all combinations of 3 instructions will function properly. That would require 521 million combinations. What about combinations of 4 instructions? Clearly instruction sensitivity studies can be taken to the extreme. However, if Motorola was that prudent the world would still be waiting for the first MC6800. The ultimate test of functionality must be thousands of customer applications.

The last issue that you mentioned related to the 6809 was the delivery of older mask sets to customers even today. I can assure you that this is not a "marketing ploy" but rather a result of our method of distribution.

Since we turn our MC6809 inventory many times a year, it is extremely unlikely that any of the older mask sets are being shipped from the factory at this time. We have much less control over devices sitting on distributor shelves. Motorola does have a policy that allows distributors to return normal excess inventories that remain unsold after 6 months. Any distributor opting to take advantage of this policy could theoretically have all of the older MC6809's off of their shelves by now.

To conclude this issue, it is indeed in our mutual interest to inform all MC6809 users of the MRDY issue. Furthermore, I recommend that you subscribe to our new literature mailing list so that you have the earliest exposure to revisions as they are issued (address attached).

From the tone of your letter, you appeared sold on our products and less sold on our integrity as a vendor. I hope that I have convinced you that we take our responsibility to our customers as seriously as you do.

Sincerely,

Murray A. Goldman
Vice President & General Manager
Microprocessor Division

MAG/ab

Attachments

October 26, 1982

'68' Micro Journal
5900 Cassandra Smith
P.O. Box 849
Hixson, TN 37343

Dear Don:

First off, many thanks for your recent correspondence concerning possible problems with the 6809 processor. As you know, Southwest Technical Products received one of the first 6809's made available outside of Motorola, and was the first vendor to place a 6809-based microcomputer on the market. For this reason, we have had extensive experience with early 6809 masks and their associated foibles.

The particular problem you describe was brought to our attention in December, 1979, by Terry Ritter, then of Motorola. The 6809 processor can lock up if the MRDY input line is found to be in its linear range coincident with a transition of the 6809 internal clock (4 or 8 MHz). The lockup problem is seen in 6809 processors with mask numbers G7F, T5A, P6F, and T6M. Note that the W8L, CW3, and GF7 (Scotland) masks do not display this failure mode. In addition, none of the external clock devices will have this problem.

It is important to realize that the lockup happens only if the MRDY input is passing through its linear range during the time that the internal 6809 clock is making its transition. If these criteria are met, then there is a non-zero probability that the 6809 processor will enter an undefined state and malfunction. The solution finally published by Motorola synchronizes the internal fast clock (now supplied externally) with the MRDY transitions.

In point of fact, both the problem and its solution are much overstated. In many applications (including all of those manufactured by SWTPC) the MRDY signals are implicitly tied to the E and Q clocks and through them to the internal fast clock. Hence it is possible to assure that the MRDY transition avoids the 6809 failure zone without modification to the processor board itself. For example, our DMF2 disk controller uses a 9602 precision timer to generate the MRDY signal on disk transfers. This 9602 is triggered early in the address cycle and is subsequently retriggered at the E clock transition. The trailing edge of the MDRY signal is centered between two of the 6809 failure windows hence the lockup problem is avoided.

The above paragraphs reflect the current state of affairs. In our dealings with Motorola, we were fortunate to have the assistance of Terry Ritter, as Motorola Management seemed determined to tell everyone (SWTPC included) that there were no problems with the 6809. Notice that the 6809 data sheet was not amended until after production of the W8L mask series had been initiated. The following paragraphs should serve to inform your readers of the evolution of SWTPC equipment, and of what sort of problems were encountered.

The initial 6809 samples used by SWTPC were packaged in ceramic and had NO identification so it is impossible to say what mask it might have been. The first parts made available in quantity were XC6809's that bore the mask number 7F (presumably pre-release G7F masks). None of these parts were shipped to customers. Those systems shipped in 1978 and 1979 used the following 6809 masks: G7F, T5A, and P6F. All of these systems were shipped as 1MHz units. Of those systems, only the ones using the SMS3509 memory unit along with the DMF2 disk unit utilized stretched cycles. If these DMF2 units are performing within spec, no MRDY failures should be seen.

In late 1979, SWTPC began investigating 2MHz systems. At this time, no 2MHz parts were available from Motorola, so parts were graded internally. These pseudo-2MHz parts used by the engineering department showed a high failure rate that was attributed to not having "legitimate" 2MHz parts. We became aware of the MRDY problem in December, 1979, and began looking into how it might effect our product. As a result of this investigation, revised alignment procedures were issued for the DMF2 disk controller. These updated procedures were designed to insure that the DMF2 MRDY signal transition did not occur in the 6809's failure window.

In second quarter 1980, SWTPC began to receive 2MHz samples from Motorola. These parts came from the T&M mask set, and were clearly marked "SAMPLE". None of these parts were shipped to customers. The first 2MHz units to be shipped came from this mask set. When customers began receiving the 2MHz units, we noticed a significant increase in the number of "mysterious" system failures. A major engineering effort was undertaken to pin down the cause of these problems. Most of these failures were traced to noise problems on the SS-50C bus and the processor board tranceiver circuit.

Application Note 122A was issued Nov 14, 1980, and essentially corrects these problems. Among the changes made was the modification of the MRDY input circuit to use a 74LS241 buffer to decrease the signal rise time into the 6809. This change, along with the revised DMF2 alignment procedure, served to minimize the impact of the 6809 MRDY failure on our product line. Once Motorola started shipping the W8L mask parts (in mid-1980), the MRDY problem appeared to have been permanently solved.

On a final, personal note, I remain skeptical of "magic" fixes to systems by replacing the 6809 processor chip. Obtaining a new mask version certainly won't hurt anything, as noise margins and address out timings have been greatly improved with the newer parts. My own home system has been using a T&M part running at 2.5 MHz since 1980 without problems. (By the way, this is not recommended for the faint of heart!) I hope I have been of some help to both you and your readers.

Norman L. Reitzel, Jr.
Software Engineer

Editor's Note: In the above letters received from Mr. Goldman of Motorola and Mr. Reitzel of Southwest Technical Products some valid points are made. And I thank them both for responding.

It is still my sincere belief that Motorola could have devised a more satisfactory manner of communicating these problems to their users. I personally know that this has caused untold hours of needless labor and probably many, many dollars spent, attempting to find that elusive 'bug' that caused our computers to 'bomb'.

As Mr. Reitzel (SWTPC) outlined in his letter, the 'magic' of resolving all our problems by a change of the 6809, is not the total answer. Yet I have received both letters and telephone calls from many users saying that the change had cured their computer ills. Maybe some or even most of them were having other problems, not related to a 'bad' 6809, still it made a difference. As to the W8L mask being a complete cure, I have received conflicting reports from individuals that I consider knowledgeable. For about $15.00 I don't want to take the risk. Once bitten is enough!

One thing is certain, if you are running any part of your system that requires a time extension of the 'E' or 'Q' signals, and you do not have either a W8L, GF7 or CW3 mask designation, or have not accomplished the Motorola 'official' fix, then I would certainly recommend the change.

Mr. Goldman certainly makes a very valid point concerning the difficulty of catching and repairing a 'bug' of this nature. I was told by some very fine Motorola engineers that they had to test the test unit, as well, in order to pin down the exact problem. And the fact, that if ALL possible combinations of CPU exercise had been attempted, we would still be waiting for the first 6800, is well taken.

It is quite possible that 'a lot to do over nothing' has come of all this, but I believe not. As for me it has been an 'education' as well as an eye-opener to some problems we were having. Fact is, I believe that a lack of multi-level communications, has been the major culprit.

When I first conceived 68 Micro Journal some five years ago, it is just this sort of thing I had in mind. One fellow called, from overseas to tell me that he has been a subscriber since issue one. He said, "If I had never found useful any other item of information, from 68 Micro Journal, this particular bit of info has cured a problem I have been plagued by for a couple of years, and has made 68 Micro Journal worth far more than we have paid since you started."

I appreciate the offer from Mr. Goldman to furnish us with more technical information, that we might pass it along to you. As I read over 70-80 magazines a month it may be somewhat difficult to screen it all, but we will try. **If you are aware of something like this (as some were) please let me know!** In order that I might pass it on to all our readers, and maybe save us all a lot of time and effort.

Concerning the SWTPC 'Applications Notes' which are not normally available to the end user, I hope to have available, in the next few months, copies available for distribution to our readers. As there are many of you who have systems that are a few years old, and because there have been numerous updates and modifications, I know that the information they contain will be useful to many of our readers. I will publish prices and availability in the next issue or so.

My thanks to all who have called or written. And despite my concern for the way that this was handled, I still want all of you to know I believe that of all the microprocessor CPUs available, **The MOTOROLA 68XXXX series is the finest.**

DMW - - -

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

INSTRUCTION SETS

Recently, I've been corresponding with John Slater of Mt. Pleasant, Mi. John wrote to tell me that he has a 6800 system, and has no plans to update to a 6809, but rather to wait for the next generation processors. Meanwhile, he would like to convert some of the published 6809 programs in '68' to 6800. John sent me a run of his 6800 assembler on some of the programs that he had typed in, but didn't know how to get converted. He had correctly changed the addresses for FLEX routines from $CXXX to $AXXX and from $DXXX to $BXXX. At that point, he tried the assembler, and scratched his head over the error messages that came out. I decided that the whole subject is worth some space here, so here goes.

Since one may take any 6800 source listing and assemble it with the 6809 assembler without any problems, why is it hard to do the reverse? The main reason is that the 6809 contains not only all the registers that the 6800 has, but several extra ones as well. That is, it can do all the 6800 instructions (almost) but it can do many more as well. When you assemble 6800 code to run on the 6809, you simply don't use its extra features. There are a few 6800 instructions that the 6809 doesn't have, but the assemblers substitute the appropriate code to do the same thing. Specifically, for example, the 6809 doesn't have the INX instruction. The assembler substitutes the code for LEAX 1,X which does the same thing.

Further, the extra registers of the 6809 have been used to great advantage. The 6809 has two index registers, X and Y, rather than just the X index

register of the 6800. It also has two stack pointers, U and S. The S pointer does what the S register in the 6800 does, but it may also be used in indexed instructions such as LDA 3,S. You can do the equivalent in 6800 code, by using the X register:

```
TSX
LDAA 3,X
```

The problem in "reverse translating" a program written to take advantage of the 6809, will be mainly, the elimination of instructions that use Y or U. The 6809 has provided the capability of putting the A and B accumulators together to form a "double" accumulator "D". Translating instructions that use the D register is, however, rather straightforward.

```
  6809          6800

LDD #$1234      LDAA #$12
                LDAB #$34

LDD COUNT       LDAA COUNT
                LDAB COUNT+1

LDD 0,X         LDAA 0,X
                LDAB 1,X

PSHS D          PSHB
                PSHA

PULS D          PULA
                PULB
```

The autoincrement and decrement instructions are easily simulated in the 6800.

```
LDA ,X+         LDAA 0,X
                INX

LDD ,X++        LDAA 0,X
                LDAB 1,X
                INX
                INX

STA ,-X         DEX
                STAA 0,X

LEAX 3,X        INX
                INX
                INX

LEAX -2,X       DEX
                DEX
```

There are a few instructions that are different in mnemonic form such as:

```
LDA             LDAA
LDB             LDAB
STA             STAA
STB             STAB

CMPX            CPX  watch this one
```

There is also a problem with the 6809 long branch instructions. The problem is the same one as fixing a branch out of range with the 6800. Essentially, you reverse the test, (use the complementary test) and branch around a jump to the label that is too far away for a branch instruction.

```
LBEQ SOMWHR     BNE SKIP
                JMP SOMWHR
          SKIP .....
```

The trickier conversions arise when the 6809 code uses both the X and Y registers, for example. What must be done is to make the X register serve "double duty". That is, it must be used for both X and Y. The trick is to save the current values of X and Y in memory locations that we could call XVALUE and YVALUE. These are two bytes long, since X is a 16 bit register. If you want a foolproof translation from '09 back to '00, just follow the following rules of substitution.

Declare the memory storage locations

```
        XVALUE RMB 2
        YVALUE RMB 2

LDY #$1234      LDX #$1234
                STX YVALUE

LDX #$1234      LDX #$1234
                STX XVALUE

LDA 0,Y         LDX YVALUE
                LDAA 0,X

LDA ,Y+         LDX YVALUE
                LDAA 0,X
                INX
                STX YVALUE

LDA ,X+         SAME WITH XVALUE

LEAY 2,Y        LDX YVALUE
                INX
                INX
                STX YVALUE

    BETTER SUBSTITUTE FOR INDEXING ON S

LDA 3,S         STX XVALUE
                TSX
                LDAA 3,X
                LDX XVALUE
```

An example might help here. The above list is not complete, but will serve as a pattern. Suppose we have a move routine to move COUNT bytes of data from SOURCE in memory to DEST in memory.

6809 code

```
  LDB  #COUNT
  LDX  #SOURCE
  LDY  #DEST
LP LDA ,X+
  STA ,Y+
  DECB
  BNE LP
```

6800 code

```
  LDAB #COUNT
  LDX #SOURCE
  STX XVALUE
  LDX #DEST
  STX YVALUE
LP LDX XVALUE
  LDAA 0,X
  INX
  STX XVALUE
  LDX YVALUE
  STAA 0,X
  INX
  STX YVALUE
  DECB
  BNE LP
```

One of the most "awkward" things about the 6800 is the fact that it only has one index register. The above code shows how much easier it is to move data in memory if two index registers are available. If substitution rules turn you off, the idea here is always to keep a copy of the present value of X and Y in the memory locations XVALUE and YVALUE, and always to go get that value when

an instruction involving X or Y is to be executed. Obviously, you don't have to get the old value of Y in X before a LDY instruction, which would change that value. The important point is that X can serve double duty as both X and Y with a little manipulation.

There is a group of 6809 instructions to transfer information between registers. These may be simulated as follows:

```
TFR X,D          STX XVALUE
                 LDAA XVALUE
                 LDAB XVALUE+1

TFR D,X          STAA XVALUE
                 STAB XVALUE+1
                 LDX XVALUE

                    OR
                 PSHB
                 PSHA
                 TSX
                 LDX 0,X
                 INS
                 INS

TFR X,Y          LDX XVALUE
                 STX YVALUE

EXG X,Y          LDAA YVALUE
                 LDAB YVALUE+1
                 LDX XVALUE
                 STX YVALUE
                 STAA XVALUE
                 STAB XVALUE+1
```

NOTE THAT THIS LOSES THE VALUES IN A AND B, WHICH COULD BE SAVED TO MEMORY AND LATER RECOVERED.

```
TFR S,X          TSX
TFR X,S          TXS
```

Now you say, how about indirect addressing? Same old approach of use X works there too.

```
LDA [POINTER]      STX XVALUE
                   LDX POINTER
                   LDAA 0,X
                   LDX XVALUE

LDA [0,X]          STX XVALUE
                   LDX 0,X
                   LDAA 0,X
                   LDX XVALUE
```

There are also register offset indexed instructions that might be the hardest to simulate.

```
LDA B,X            STX XVALUE
              LOOP INX
                   DECB
                   BNE LOOP
                   LDAA 0,X
                   LDX XVALUE

                     OR

                   STX XVALUE  SAVE VALUE OF X
                   STX SCRATCH
                   CLRA
                   ADDB SCRATCH+1
                   STAB SCRATCH+1
                   ADCA SCRATCH
                   STAA SCRATCH
                   LDX SCRATCH
                   LDAA 0,X
                   LDX XVALUE  RESTORE X
```

There are a few more peculiarities in the conversion. The 6809 doesn't have explicit instructions for the status register such as CLC, SEC, CLV, SEV. The carry is cleared by the instruction ANDCC #$FE or ANDCC #%11111110. The carry is the low order bit of the CC register. SEC would be done by ORCC #%00000001. CLV = ANDCC #%11111101, and SEV = ORCC #%00000010. These may be memorized or figured out from the instruction set summary sheet. These instructions do not occur frequently in assembler code.

I've saved the absolute worst till last. What if the 6809 code uses the User stack? If the use is infrequent, perhaps you can see a simple way to use some fixed memory location. If it is extensive, you will have to use X again.

```
LDU #$BE00       LDX #$BE00
                 STX UVALUE

PSHU A           LDX UVALUE
                 STA 0,X
                 DEX
                 STX UVALUE

PULU B           LDX UVALUE
                 LDAB 0,X
                 INX
                 STX UVALUE
```

This list is by no means complete, but most of the problems have at least been touched upon, and suggestions made for substitute code. Now, having waded through most of the problems I have a suggestion for some of you 6800 owners. Does one of you happen to have Hemenway's 6800 Macro Assembler? Why not write a macro file that would do the above substitutions. You could come up with a 6800 assembler that would accept 6809 code and assemble 6800 object code. You don't have a 6800 Macro Assembler? Write a program in BASIC (Pascal, C, or whatever would do too, but TSC BASIC is rather universal to most of the 68XX users) to look for 6809 instructions and make the substitutions by reading the 6809 assembler source file and writing a 6800 assembler source file. It will be slow as molasses but who cares. Take the 6809 source code and add the necessary RMB's:

```
XVALUE RMB 2
YVALUE RMB 2
UVALUE RMB 2
```

Now read the source file with a program that searches one line of the source code at a time for all the 6809 only codes, making the appropriate substitutions in the output file. Best approach would probably be to add a '*' to make the original instruction a comment, and then add the substitute code. When you have included all the 6809 only instructions in your program, with the appropriate "universal" substitutions, you will have a 6809 to 6800 translator.

There are some sticky parts to this. PSHU may be followed by a parameter list, PSHU A,B,X,Y etc. You must study the 6809 instruction card to be sure you always stack in the correct order.

If anyone cares to tackle this project, I'm available to get you started, help debug, work on the trickier parts, or whatever. Sounds like a good group project to me. If we could come up with a translator that would work on 90% of the instructions, it would handle most of the utilities that are "public domain". Of course if you are willing to buy 6809 programs and run a 6809 disassembler on them, you could use a translator to make them 6800 compatible!

I'm by no means an expert on the two instruction sets, and I haven't really given this problem a great deal of thought. I have no doubt that there may be more efficient substitutions that will work universally, but I suggest that simplicity is better than saving a few bytes of code. Some of the code, such as the register offset indexed instructions would benefit by a subroutine such as ADDBX and ADDAX, rather than substituting inline

code each time such an instruction is used. This could be accomplished by a translator program also.

Well, are there any experienced programmers out there, or perhaps someone willing to struggle a bit to try this project? John Slater calls himself a neophyte. I'm sure that there are many more in the same situation. If we can get our heads together and do the job, I will publish the results here.

NEWS ITEMS

Last time, I mentioned A/BASIC. I now have a final copy that has all my reported bugs removed. I have compiled a bunch of my old programs that I had done for the 6800 version, and all now compile with no changes, and run as expected. Perhaps some of you had or have the 6800 version of A/BASIC. I should mention that there are a few enhancements in the '09 version. The best of those is the fact that IF THEN statements now allow a statement rather than just a line number. That is, the old version was limited to IF A=B THEN 200. The new version has an ELSE added to the capability, and allows a statement, as in IF A/2 * 2 = A THEN PRINT "EVEN" ELSE PRINT "ODD". This is a major help in writing more structured programs. A/BASIC is available from Frank Hogg Laboratories. See their ad.

COLOR User Notes

ROBERT L. NAY
4429 Plantation Lane
Norcross, GA 30071

HELP! HELP! What are YOU doing with the Color Computer? I realize that many of you are still 'getting the feel' of Computers with your System, but there are many others who have found a serious use for this Computer. Let us know what you are doing with this System, so we can pass it on. I would prefer it to be in the form of an article (if you have any questions as to form and content, drop me a line), but pass on your ideas for others to build on. I'll pass on some of my thoughts, and throw out a few ideas, and hope that I can generate a little 'User Activity' out there!

I have been concentrating on Reviews and FLEX in for the last Year or so, for a couple of reasons. As normally happens when a "New" System hits the Market, a lot of people see a way to make a 'fast buck' and hurry new Software out to get on the Market as soon as possible. I'm sorry to say that a lot of it is "JUNK", and should never have been offered for sale. I felt that it was important to let you know what we had received that was worth spending your money on, and also hoped that they would be somewhat 'instructional' to the new Computer users. I'm sorry to say that we at '68' Micro have had to turn down several potential advertisers, and I have not reported on some Software, because it did not meet our Standards (which are simply that the Advertisement must 'PROPERLY represent' the product). I am also HAPPY to state that the above trend is rapidly slowing down, and most of what we receive now are really GOOD Products. I will continue to do my best to objectively represent that which we receive as soon as possible after it arrives, because we feel that this is one of our responsibilities to YOU. I'll discuss the FLEX a little later.

I see the Radio Shack TRS-80C Color Computer as filling three major 'gaps' in the Microcomputer Industry. First, it is potentially one of the most powerful Computer Systems available today, primarily because of the 6809E CPU under the Top Cover. Add the price to this capability, and you have a System that is so far beyond anything else in 'value received for dollar spent' that it has no competition, Sure, this would be nice, and that would be nice, but where can you get a Computer System for $400 which is ALREADY capable of operating with 96K of Memory; that contains the most powerful 8-bit CPU available today (with MORE 16-bit power than a couple of the so called 16-bit Chips); that can be expanded from 16K to 64K RAM for less than $100 (I see Apple Cards advertised all over the place for adding 16K to that System for the same price); and that can be brought up on a MAJOR Disk Operating System, with all of the Software that has been developed for it over the past five years, for the price of the Disk Operating System and some RAM. And with all of this, you have an Excellent and Powerful BASIC Operating System, a 'consistent and 'open' Memory Map, and an accessible and fairly complete Expansion Port, all of which lead to the ease of expansion, programming, updating, etc,, etc. And with COLOR, even!! WOW!!!

Second, the Color Computer makes an OUTSTANDING Educational Tool. This area has not even been scratched. I understand that Radio Shack has some excellent Educational Software 'on the way'; I know that they have PILOT and LOGO about ready to go, and hear rumors of other products. Let's examine this potential a little more.

Video Tape Instructional material is becoming more popular, so educational institutions either already have Color TV Sets, or will shortly be acquiring them, so that, in many cases, they will not need to purchase additional sets for the Computer. Or, if they do, then the Sets are available for the Tapes. This leads to a lower investment for a greater return, and more efficient use of the available resources. Panasonic now has a system which allows random access to 199 sections of Material on a Video Tape; it has some severe limitations, but it is a beginning. This concept provides even more potential for an Integrated Educational System (it is a simple matter to add the Computers' output to the material already on the Tape; the possibilities are increasing exponentially.). Also, there are a few RS-232 controlled random access Slide systems available, which increases the possibilities even more. With the Color Computer, you have an interactive educational capability that is so far beyond a 'lecture' that the amount of knowledge gained in a given amount of time can be increased astronomically.

The combinations of Color and Sound available also drastically increases attention span and interest. With proper programming, a dull spelling lesson becomes an interesting and challenging GAME. Not only that, but Testing and Test Analysis can be incorporated within the Program, freeing the teacher from the hours of Grading and Recording Tests, and allowing more time for Lesson Planning, constructing Specialized Lessons, Developing new Programs, etc. Or, instead of a spelling lesson, how about a Graphic Demonstration of Electron and Hole flow through a Semiconductor, or the flow of an Instruction Cycle through a Microprocessor Chip, or an Industrial Process, Chemical Process, etc., etc., etc. A properly constructed Program would not just indicate that an answer was incorrect, but would determine WHY it was incorrect, and LEAD the student back to the correct answer. The Computer can provide a truly Modular Lesson Path, and through the Discovery Learning Process, providing a consistent material coverage which advances at the individuals optimum rate, and with an efficiency which is MAGNITUDES BEYOND any Educational System in use at the present time.

But it Costs too much! BAH and HUMBUG!!! The real COST in the present day Educational System is the WASTE and INEFFICIENCY. Colleges are struggling to find methods for providing all of the material needed for a student to graduate with a decent FOUNDATION in a given field, let alone with the

knowledge required to be of minimal value to an employer. They have to spend the first two years of College teaching the three R's, and then HOPE the student can learn to think WHILE he is trying to learn the material industry demands that a graduate have. The High Schools are so shackled with the 'discrimination problems, and I don't mean RACIAL, that they have to hold back the 'college materiel' students to the pace of slower learners. And this filters all the way back thru Grade School. So they strike a happy (???) median, and hold some back, and rush others, so that our present Education System produces a majority of students who are either bored stiff, or have been totally by-passed 'by it all'. The Computer, properly utilized, could allow each student to proceed at his own pace, while still assuring that a SPECIFIED minimum level of knowledge be reached. Colleges and Vocational programs could concentrate on their specific areas, and KNOW what the minimum level of the entering students would be. No, the COST of Computerizing the Educational System would be NOTHING compared to the waste that would be saved.

Well, enough "Soap Box"; this should stir up some discussion. Throw rocks, or throw flowers, but let us hear from you.

The Third 'gap' that the Color Computer is filling is in the area of 'Control Systems'. Again, the price and capability make this an unbeatable System for Control Applications. Many Companies are finding that they can not develop a 'Controller' for the price of a Color Computer. The CPU, Expansion Port, and System Capabilities allow Instrumentation and Control Problem Solutions without the need for Circuit Design and PC Board Layout and Production expenses. Additionally, programming is a snap, with no requirement for Software Development Systems, Simulators, Program Conversion from the Development System to the Controller, etc. I know of numerous Organizations that are using the Color Computer in these types of applications.

As I have stated in the past, '68' Micro Journal originated as a forum for the 68xx CPU Users. It has developed into a 'software oriented' magazine because the FLEX Operating System developed into the primary Disk Operating System for the 68xx Based Systems (and is extremely friendly to Software Writers), and because the Major Manufactures developed a Bus Standard that is the envy of the Microcomputer Industry (the SS-50 Bus). Also, the Hardware has been available for a while, so most of the Hardware Development has stabilized into accepted practices. The Color Computer has emerged, and should provide a fertile field for both Hardware and Software experimentation, development, and discussion.

Let's use this "forum" for what it was intended. I would like to see this develop into a Magazine for IDEAS. I can imagine several areas that could become regular "Columns"; not written by one individual, but like the 'Bit Bucket' Column. These could include Education (ideas and programs on Teaching methods with the Computer, Testing Methods, Administrative Processes such as record keeping, scheduling, etc.), Graphics (such as faster Display Algorithms, use of Colors, 2D and 3D Graphics, etc.), Hardware (such as Keyboard Improvements, A to D and D to A Converters, Voice and Sound Chip Interfacing, additional Ports, Interfacing an MMU, Timer and Baud Rate Generator Installations, Power Supply Improvements, RGB Outputs, Display Enhancement, etc.), Software (Utilization and Limitations of the BASIC Operating System, Assembly Language Utilities, use of Subroutines in the BASIC ROMs, utilization of the Hooks provided in the BASIC ROMs, etc.), and so forth. There are a few people working full time with this System, more working on it in hopes of generating some extra income, but the vast majority of the really GOOD Ideas and ADVANCEMENTS are made by the Hobbyists who has some particular problem he has solved. By sending in your idea for publication, you accomplish several things. First, you generate solutions to problems many others are working on, or give them ideas which helps them solve a problem in a different, and maybe better way. They then publish their solution, which generates more ideas, etc. Second, you are recognized by Users who may be working on a similar problem, and a flow of information begins taking place that would not have been available had they not seen your name and become aware of you. This again leads to better solutions to problems, and more ideas for other solutions to possibly unknown problems at that time. And third, it gets you in contact with more Users, and often people in your own locality that you did not know, that has a Color Computer. '68' Micro Journal is a Magazine FOR Users BY Users. Let us hear from you; share what you have gained with others, so that everyone may gain more from them, including you.

More on COLOR-FLEX Compatable Software

It seems like I have been spending the majority of these Columns on the FLEX Operating System lately. There are several reasons for this; most important, because there is a LOT of 'Working' Software (meaning both definitions of the word; they Work, and they DO Work) available which runs on the FLEX DOS. In past Columns, I have mentioned many of the Programs that are compatible with the DATA COMP FLEX Conversion for the Color Computer (and many are compatible on other Conversions, also). One of the big shortages has been a GOOD 'Screen Oriented' Editor and Word Processor. We should have the initial conversion of STYLOGRAPH Operating with this FLEX Conversion by the time you read this Column. As of right now, 'STYLO' is the only Word Processor available that supports proportional spacing on a Daisy Wheel Printer, although SCREDITOR III (available from Alford & Associates, see Advertisements) will have the capability in the near future. The problem with these Conversions is that they are very 'Display Intensive' Programs. That is, they require a LARGE amount of control over the Display to be able to provide an up-to-date layout on the Display Screen of the formats, deletions, insertions, etc., as they occur. With a 'Screen Oriented' Display, what you see is what you get. These Programs by-pass the FLEX System to work directly with the Display, and often use Control Codes that the FLEX Routines filter out during processing. This means that they require a MAJOR re-write to be able to work with the Display and Screen Format (51x24) that is used with the Color Computer. Now, this type of Software is becoming available for the Color Computer Conversions. Watch the '68' Micro Journal Advertisements for Pricing and Availability. (DATA COMP will be offering the "Color STYLO" Conversion as soon as it is ready.)

Many of the normal '68' Micro Journal Advertisers have Software that is compatable with the FLEX Conversions for the Color Computer. I have mentioned some of them in past articles, and will continue to mention others as we get the time to make sure they work OK on the system.

Many of you are familiar with the excellent Software from COMPUTER SYSTEMS CONSULTANTS, INC. Dr. Bud Pass, who owns COMPUTER SYSTEMS CONSULTANTS, INC., has written numerous magazine articles over the last several years, many of which have appeared on the pages of this Magazine. His SUPER SLEUTH Disassembler and TABULA RASA Spreadsheet Programs are known across the industry as being extremely powerful pieces of Software. SUPER SLEUTH is offered in two versions, one for the 6800 Series Computer Chips (6800, 6801, 6802, 6803, 6805, 6808, 6809, and 6502), and another version for Disassembling Code for the '80 Series Chips (8080, 8085, and Z-80) on 6800 & 6809 Systems. The flexibility and power of these Disassemblers has to be seen to be believed. TABULA RASA is somewhat

similar to the popular DESKTOP PLAN type of Electronic Spreadsheet tabular computation Software, and again, is EXTREMELY Powerful. This Program offers all types of calculations, and can be used for ANY type of tabular computations, including both Business and Engineering. Also included in his list of Software are Cross Assemblers, Debugging Simulators, a Translator system which allows translating 6502 Code to 6809 Code, a Program which is a big help in converting 6800 Programs into 6809 PIC Code, and Business Type Software such as Mailing List, Screen Formatting, Inventory, etc. Bud also sells a Program which allows UNIflex Software Developers to Debug their Programs using FLEX based Software Development Tools. Although I have not run ALL of his Software, all that I have used has operated with no problems on this FLEX Conversion. If you have a question, drop him a line. I'm sure he can tell you if there might be a problem or not, and the Software is OUTSTANDING.

REVIEWS:

COMPUTERWARE, Box 668, Encinitas, CA. (704) 436-3512 has been supporting the Color Computer from the beginning (and supporting the 6800/6809 Systems for several years), both with their own Hardware and Software, and as a supplier of other Manufactures Products. They have an excellent selection of Products for use with the Color Computer, including "working" Software (Programming Languages, Business Programs, Word Processors with Spelling Programs, Utilities, etc.), "fun" Software (Games, etc.), and all types of Hardware (Mod Kits, Printers, Serial to Parallel Converters, ROM Paks, Modems, etc.). They also have a free Newsletter which appears periodically containing new products, a Catalog, and general information about the Color Computer (drop them a note to get on the Mailing List). To give you an idea of how fast they are expanding their coverage of this Computer System, we received five (5) new Programs from them this month, including a Disk Communications Package for Modems, Terminal Operation, etc., two Graphics Programs, and two Game Programs.

The Color Connection - A Communications Package
Requires 16K RAM min., EXBASIC NOT required.
Cass -- $29.95
Disk -- $39.95

The COLOR CONNECTION is a Machine Language Program which allows the Color Computer to be used as a Smart Remote Terminal for use with other Computer Systems which use the RS-232, 300 Baud format; to be used with Modems (including "Smart Modems" with Auto Dial/Answer, etc.); and for Communications with other Computers. My ONLY complaint is that it does not provide a method for controlling the Baud Rate; it is set at 300 when the Program is executed. This is only a problem if you are using The COLOR CONNECTION on a 1200 Baud Modem or for Communication (direct hookup) with another Computer; the large majority of the time the 300 Baud is the required Baud Rate.

The COLOR CONNECTION features include Auto Dial, the use of all Free Memory for Buffer Space, Stores all or any line or lines from Buffer to Disk, it is Menu Driven, it provides for User Defined Macros (up to 40 characters per Macro), and allows loading the Buffer from either Disk Files or from the Keyboard.

The Main Menu provides the overall control of the Program. It displays the Remaining Buffer Space and allows the following controls:

CHANGE SET-UP FILE -- Brings up the SET-UP MENU which allows setting 7/8 Bit operation, Parity, Line Feed control, Full or Half Duplex, enter the Phone Number to be used with a "Smart" Modem, define four (4) different Macros of up to 40 characters each (for entering Passwords, ID Numbers, etc.), Save the Set-Up to Disk, and go back to the Main Menu.

LOAD A SET-UP FILE -- Loads a previously Saved Set-Up File.

TERMINAL FUNCTION -- This selection enters the Terminal Mode of Operation. The "Down Arrow" Key is used as a "CONTROL" Key, both for controlling the Program, and for sending 'control codes' to the host System. The Controls are as follows:

CTL 1 - Open Buffer (Cursor turns PINK)
CTL 2 - Close Buffer (Cursor turns BLUE)
CTL 3 - Clear Buffer
CTL 4 - Exec AUTO-DIAL (<BREAK> aborts)
CTL 5 - Exec Macro 1 (")
CTL 6 - Exec Macro 2 (")
CTL 7 - Exec Macro 3 (")
CTL 8 - Exec Macro 4 (")
CTL 9 - TRANSMIT Buffer (")
CTL 0 - Return to Main Menu
CTL H - Backspace
CTL I - Moves Cursor forward one space
CTL J - Moves Cursor down one line
CTL K - Moves Cursor up one line
CTL L - Clears Video Screen
CTL M - Carriage Return (same as <ENTER>)
CTL * - Takes the Hayes Smart Modem off line

You have the full Control Key set available; this list is the Key definitions for this Program.

BUFFER MENU -- Brings up the Buffer Menu which allows you to VIEW BUFFER and move around in it, Open and Close Files for storing any of the Lines, etc.; Save the Buffer to Disk; Dump a Disk File to the Buffer (the File must be in the ASCII Format); Enter Text into the Buffer from the Keyboard; and to Return to the Main Menu.

RETURN TO BASIC -- Exit the Program

The COLOR CONNECTION contains routines which keep a word from being broken at the end of a line on the Display Screen (unless it is more than 15 characters long), making the text much easier to read. It also provides Auto Repeat in both Terminal and Buffer Modes of any Key; if the Key is held down, it will be repeated automatically.

If you are used to a Modem Program such as VIDEOTEX, you will find that The COLOR CONNECTION works a little different. The Buffer capabilities allow you to save what you are seeing on the Screen for later examination and/or printing. This is a BIG help; for example, I spent a couple of hours on Compuserve the first time I dialed it up just in the Menu - with this Program, you can get it into the Buffer, save it, and GET OFF THE PHONE, then spend as long as you want studying it. You can make up a Text (ASCII) File, such as a Bulletin Board message, ahead of time; then Dial up the phone, hit a Macro for your Password and ID Number, dump the file to the Buffer, Transmit it, and get back off of the phone quickly. You will save enough money to make a lot more use of your Modem. COMPUTERWARES "COLOR SCRIBE", among many others, would make a perfect companion Program, as it makes the job of editing and storing a Text File easy, and provides an excellent method of listing a saved file from Buffer to the Screen or a Printer for study after you have gotten off of the phone.

A Program of this type is almost a necessity for accessing the College Computer Systems, for example. Normally, you are not too worried about long distance phone bills with these systems, but waiting on the 300 Baud activity gets OLD in a short time. A Program like The COLOR CONNECTION allows you to spend the hours required to make up a PASCAL Program, for example, and get it in pretty good shape before you call up the Big System. You can then put in your call, dump the Buffer to the "Big Boy", and see if it will compile and run. In my case, I have access to a CYBER at School, but the Editor and Text Processor are pretty "archaic" (interpret as "rudimentary", or "lacking in capability"), even for a "Batch" organized System.

I can do most of my work on the Color Computer "Smart Terminal" at home and dump the File into the CYBERs memory bank, then pick it back up at School through the Terminal there and use it as required. I can Edit it, etc., and still use it at home also. Nice!!!

This is an excellent Program, and should find a lot of usage by the Color Computer owners. It is easy to use and, with the one exception of Baud Rate Control, provides a good "working" piece of Software. This is more important than many of you may realize; a GOOD program is not necessarily one that provides "every possible" control, command, etc, Unless you are "living" with that program ONLY, you spend more time in the Users Manual than you spend USING the program. It gets moved back in the "stack" and used less and less, until it is "just another program that you seldom use", and probably gets replaced. Again, this is a GOOD, WORKING program, that you will be able to "get comfortable with" is less than an hour, and will find more and more use for it.

FOXYGRAF - A Graphics Development Program
Requires 16K RAM min., EXBASIC NOT required.
Cass -- $29.95
Disk -- $34.95

FOXYGRAF is a relocatable, ROMable Machine Language Program written by Bob Crispen for COMPUTERWARE which provides both an excellent method of developing Graphics and an Instructional Course on the Graphics capabilities of the Color Computer. This Program comes with an excellent Users Manual that not only explains how to use the Program, but provides a discussion of the control and operation of the Color Computer's Graphics capabilities with examples in Assembly Language, BASIC, and FORTH, It also provides some insight into some of the methodology and problems involved in accomplishing Graphics Programming. Finally, Bob goes the "extra mile" with this Program and the Manual in advancing the use of Graphics on the Color Computer by providing an extremely complete discussion of his Program and it's design philosophy, the Memory Utilization, Buffers and their use, Patch Addresses and their requirements, and the Subroutines used in the Program. In his opening remarks in this section of the Manual, after reiterating the Copyright warning, Bob states

> "My advice to users is the same as Stravinsky's advice to musicians -- don't borrow these ideas; steal them. When you borrow something, you take care to leave it as it was, When you steal something, you beat it up and generally make it your own. Every one of the subroutines in FOXYGRAF shows the effects of having to cope with a set of restraints peculiar to FOXYGRAF. Read the disassembly (the one you make - RLN) of FOXYGRAF for the purpose of understanding what happens, then if you want to use any of the routines, make them solve your own problems."

Well said, Bob. Maybe this statement should preface any Disassembler Manual written. Anyway, this gives you some idea of the "Philosophy" of Bob's Manual; there is 56 pages of GOOD INFORMATION here that makes this Program one of the "Best Buys" on the Market. It is extremely well written, with an attitude that lets you know he is "human" just like you and me, and not some "omnipotent programming GOD" that only produces "perfect" code that the User will NEVER find a reason to modify. (Why FOXYgrat?? Again, I'll let Bob explain it in his own way. From his Dedication:

> "FOXYGRAF is dedicated to my wife, Kelly, for her love and patience; to Foxy, our founder and mixed (mostly Norfolk) terrier who was always on the wrong side of a door while it was being written; ..."

And I thought that only applied to Cats!!)

FOXYGRAF operates similar to many of the Monitor Programs, with single-letter commands. It provides a Help Command, and a "Query" page (actually, two pages) of 'status' type of information. The Commands include "A" for setting the Screen Display page, "B" for Block Move, "C" for specifying a Color, "D" for entering the Draw Mode, "E" for drawing an Ellipse, "F" for Filling an area of Memory, "G" for Go Load from Tape, "H" for Help, "I" for Initializing the Program, "J" for Jump to subroutine, "K" for setting a 'Kolor' in binary, "L" for controlling the Tape Load request from a "G" Command above, "M" for setting the VDG Mode, "N" for a New Cursor or Background Color, "P" for setting the Display Screen via Page control, "Q" for going to the Query Pages, "R" for Returning to the graphics mode and page you were in from a Help, Query, Load or Save command, "S" for Saving a block of Memory (Graphics), "T" for moving the Cursor back to the Top, "W" for setting the Cursor Motion speed by changing the Wait time, "X" for eXiting the Program, "Z" for Zero - hiding the Cursor till it is moved, "/" for specifying an end of a line, and <BREAK> which aborts any command in progress. Many of these Commands expect further information, and this summary does not provide the full meaning of many of them. The missing letters, O, U, V, and Y, are not used, but provision is incorporated in the Command Table for you to add them with your own routines.

Space does not allow room to be able to give you any realistic idea of the capabilities of this Program. For example, one of the things you might do is create animation by Drawing a picture, Block moving it to another Screen and making small changes, and after you have set up several Screens (notice you don't have to completely redraw each one), write a little routine to rapidly Page through them to see the motion. One capability discussed is the use of the Mass Storage System (Tape or Disk) to save different 'pictures' and then view them as a 'Slide Show' with single-key commands.

If you are interested in the Graphics Capability of the Color Computer, FOXYGRAF is a MUST. Bob has laid the groundwork (and the Program is presented as just that), but this is just scratching the surface. Even if you value your time at less than a penny and hour, you can't duplicate this work for any $35, so why not order a copy of FOXYGRAF and carry on from there? Bob would be GLAD to hear of any extensions you make to it, and '68' Micro Journal would be happy to publish any ideas you have along these lines also. Things like more efficient algorithms for line drawing routines, circle routines, 2 and 3 dimensional presentations, etc., etc., etc. We have "not yet begun to fight" with this machine, so to speak. Bob has taken the first step; lets' carry on from there.

SEMI DRAW - A Graphics 'Sketching' Program
Requires 32K RAM AND EXtended BASIC
Cass -- $21.95

COMPUTERWARE's SEMI DRAW is a BASIC Program which makes use of numerous Machine Language Routines that use the 3 Graphics Modes which are not accessible through the normal BASIC or EXBASIC Commands. These are the Semigraphics-8, Semigraphics-12, and Semigraphics-24 Modes available when the Motorola MC68883 SAM Chip is interfaced with the MC6847 VDG Chip, as is used in the Color Computer. These Modes are called 'Semigraphics' because you can INCLUDE Alphanumerics with the Graphics. The 3 Modes provide 64 'elements' across the Display Screen and 64, 96, or 192 lines vertically. If you consider each normal Alphanumeric Character 'block', you can see that these Modes break each character 'block' into 2x4 'boxes' for Semigraphics-8, 2x6 'boxes' for Semigraphics-12, and 2x12 'boxes' for Semigraphics-24. (See the Motorola Data Sheet for the SN74LS783/MC6883 for details,) You have the use of all 8 Colors in these Modes, but there can be some interaction between them.

SEMI DRAW allows you to use either the Joysticks or the Keyboard for drawing your pictures. You can use 'Single Key' Commands to control various functions, such as 'setting' a point to the chosen Color, changing Colors, etc. Some of the interesting capabilities they provide are the ability to Repeat (Copy) a picture on another Screen (which allows you to make small changes for animation), automatically Page between the pages (to see the motion), and copy the pictures to Tape or dump a Screen to a Line Printer VII, Line Printer VIII, or a NEC 8023 Printer.

Again, if you are interested in Graphics, this Program is well worth the money. Since it is primarily written in BASIC, you can study the program for programming ideas, as well as investigate the other three Graphics Modes. COMPUTERWARE also provides three 'Pictures' to show some of the capabilities of these Modes, two of which are animations.

DOODLE BUG -- Game
Requires 16K RAM, EXBASIC NOT required.
Cass -- $24.95
Disk -- $29.95

RAIL RUNNER - Game
Requires 16K RAM, EXBASIC NOT required.
Cass -- $21.95
Disk -- $26.95

These two COMPUTERWARE Games follow their tradition of providing Colorful, challenging Games which are well supported with Sound. I feel the 4-Color Graphics will become the 'Standard' for most Games offered for the TRS-80C Color Computer. The 2-Color Modes offer finer resolution, but lack the variety and 'vitality' available through the use of more Colors. COMPUTERWARE, along with some of the other Software Producers, seem to have discovered this factor earlier than most. Developing a GOOD Game, through the proper merging of the Color and Sound capabilities of the Color Computer, has been a challenge, and it is heartening to see that some 'artists' are beginning to get the feel of it. Some excellent examples include, but are no means limited too, these two Games, their Color Pak Attack, Mark Data's BERSERK, and Radio Shack's POLARIS. These are examples of what CAN be done with the MC6847 VDG - MC6883 SAM combination. Each provides a Colorful, challenging, game which does not 'get old' from continuous playing.

RAIL RUNNER is a program which requires timing and fore-thought. It is somewhat similar to the FROG and CHICKEN Type of Game in that the object is to cross several Rail Road Tracks, pick up a Hobo, and return, all without getting run over. Things can get pretty 'hairy'. This will be a good game for the 'younger folks', and is interesting for the adults.

DOODLE BUG could end up being a Classic. It is somewhat similar to the PACMAN games in that you are maneuvering within a maze, 'eating' various and asundry items to gain points, while staying away from the 'baddies'. BUT, the 'kicker' in this game is that you can duck THROUGH the walls to escape the 'baddies'. Not only that, but the wall section pivots, closing the path you just came through. The 'baddies' can NOT pass through the wall. With proper planning, you could possibly pen them up so they could not get too you (I think - It's real time action and you have to think faster than I have been able too so far). This Game has excellent Graphics and Sound, and a depth of complexity approaching Chess. If you are a Game Fan, DOODLE BUG is a MUST.

CHEAP TALKER

The Radio Shack Color Computer is truly a remarkable machine for the money, and I beleive that there are quite a few Color Computer users like myself that are operating on a very tight budget (still saving for that disk drive). But with all of the recent articles on speech synthesis, I couldn't wait any longer for my Color Computer to talk! This article describes the "Cheap Talker" (as opposed to the not cheap enough ,but very nice, Sweet Talker from Micromint Inc.) that I built and programmed in one afternoon for my Color Computer.

The Cheap Talker requires only 2 IC's and a transistor with a few resistors and capacitors placed on a edge connector type circuit board that Plugs into the Color Computer cartridge slot. Certainly, a Printed circuit would be nice, but I wired mine Point-to-Point (using sockets for the IC's) and placed the finished board in a modified 8-track cartridge (as suggested by other articles in the Past). The software is a simple Basic Program and is stored on cassette. Just Plug in the cartridge, turn on the computer, load the Program, and listen to your TV say "I am the Color Computer Talker" followed by the ABC's. Needless to say, the applications are many, as my not quite 2-year old daughter tries to recite the ABC's along with the computer (a homemade SPeak and SPell, SPeak and Math, SPeak and Read, etc., are just a Program away).

The 2 IC's used are a Motorola 6821 PIA (Peripheral Interface Adapter) and the Votrax SC-01 SPeech Synthesizer. Although the circuit board that I used is made by Vector (3719-1), Radio Shack is soon to have an experimenter type board for the Color Computer which might be cheaper. The Vector board also must be cut down to fit into the cartridge door and the edge of the Color Computer. Total cost of the Project was less then $25.00 Plus the cost of the Votrax chip (I bought the SC-01 from Quest Electronics for 39.00).

Two Motorola 6821 PIA's are used in the Color Computer for the Keyboard, I/O, and other functions. Just about any PIA device may be used (6522, 8255, etc.), but since the 6821 is available in the Color Computer, I thought most users might be more familiar with its Programming and operation. I chose to use port A and the CA1 and CA2 control lines to drive the SC-01 chip since I plan to connect a General Instruments Programmable Sound Generator to Port B later.

The Votrax SC-01 SPeech Synthesizer is a 22-Pin CMOS IC and is Powered with 12 VDC in this circuit. SPeech is synthesized using Phonemes to build words. The SC-01 has 64 different Phonemes (including Stop and no sounds) that vary in duration as shown in the Phoneme chart. Thus a 6-bit code defines the desired Phoneme and the timing and sound are Provided by the SC-01. The Pitch of the voice may be varied by changing the master clock frequency (with a Potentiometer) or with inflection inputs. I chose to ground the two inflection inputs but they may be easily added and should be buffered with transistors or TTL 7416 high-voltage open-collector circuits as shown. (I was able to drive the I1 and I2 inputs directly from the 6821 PIA but Votrax suggests these inputs be .8 x VP so they are really not TTL compatible). The data lines are compatible with 5V inputs and are driven directly by the 6821 PIA.

The two control lines from the 6821 PIA Provide the necessary handshake with the SC-01 for continuous speech. The Strobe (STB) is a 5 Volt compatible input that latches the Phoneme 6-bit data code. Latching occurs on the rising edge of the strobe signal. The Acknowledge/Request (A/R) is essentially a CMOS level output and is buffered with a simple transistor circuit. When this A/R signal goes from low to high (6821 input goes from high to low due to transistor inversion), the old Phoneme has timed out and a new Phoneme data code may be latched into the SC-01.

The Audio output is fed through the cartridge sound Pin (35) to the Color Computer and out to your TV. The sound multiplexer IC in the Color Computer is selected during the Program initialization. The output voltage from the SC-01 should be a maximum of approximately 3 VP-P for the AM Phoneme and is sufficiently large enough for Good volume (an amplifier and separate volume control might be easily added, but be careful to limit the Color Computer sound input to about 5VP-P maximum).

The Program selects the cartridge sound input, sets up the PIA, and then outputs a Stop code (63). Then the sign-on message is read (and spoken!) leaving the user to create speech with Phonemes, separated by commas or spaces, in a string. Since this Program is intended only for demonstration and experimentation, a string of Phonemes should be long enough to say a few words and test the "Cheap Talker". Vary the frequency control to change the voice Pitch and if you connected the inflection inputs, IN0 through IN3 will add the Proper values so that Port A will also output these codes.

John R.Kelly
1440 N 61st
Lincoln, NE 68505

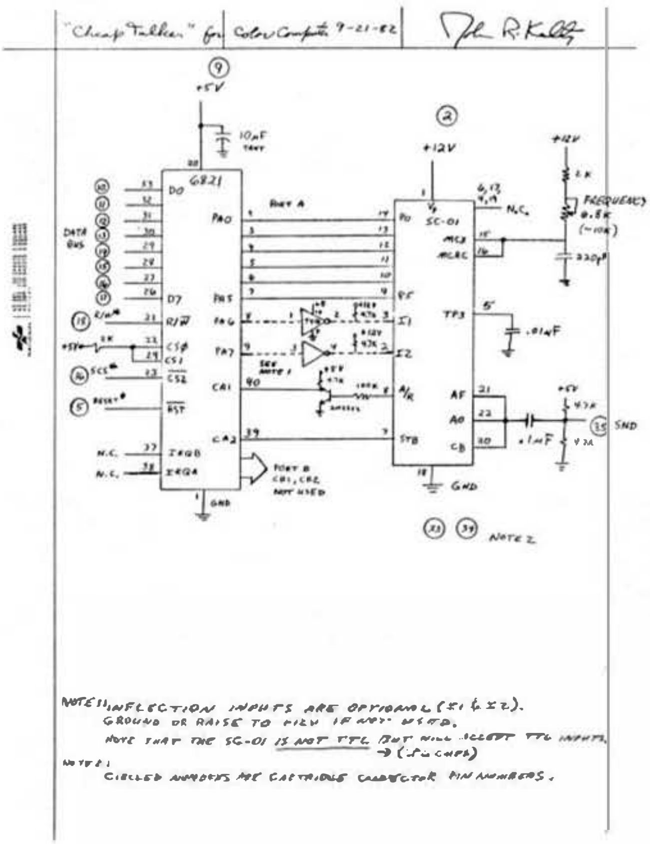
ADDENDUM

Looking back over the notes (schematics, article), some references should be made at least as far as inspiration. I had originally read and built my own version of "Voice Synthesis for the Color Computer" (Byte Publications, February 1982). This inspired me to build a software phoneme generator (which I did, using my own voice to develop the phonemes). It worked, but there are many problems (stuttering, gargling, uneven volume, etc.).

Then a friend purchased the "Micromint "Sweet Talker." After looking at the documentation, which was extremely good, I decided to build my own Talker. I waited until someone (namely Quest Electronics, 2322 Walsh Ave, Santa Clara, CA 95051) would sell the Votrax SC-01 circuit for less then $75.00 as advertised earlier by Micromint Inc.(917 Midway, Woodmore, N.Y. 11598).

William C. Clements, Jr. had an article in the June 82 MICRO (the 6502/6809 Journal) on how to "Add a VIA and Speech Synthesizer to the Color Computer", but this used the 6522 VIA and Sweet Talker. His program would load codes into a matrix and say words and I like to type in phonemes better. I plan to build a talking bulletin board and to add the General Instruments AY-3-8910 Programmable Sound Generator soon.

I have enclosed copies from the Votrax specification sheets, the Motorola Microprocessors Data Manual (1981), and the Radio Shack TRS-80 Color Computer Technical Reference Manual (Cat. No. 26-3193). (These last two manuals are always nearby my Color Computer!)




```
1 'COLOR COMPUTER CHEAP TALKER
2 'JOHN R. KELTY
3 '1440 N 61ST LINCOLN,NE 68505
4 '(402)467-3290 HOME
5 '(402)472-2793 WORK
6 'SEPT 23, 1982
10 DIM Z(200):A=65344:CLS
11 IN=0 'INITIAL INFLECTION
12 'SET UP 6821 PIA
13 'PORT A, PA0-PA5, PHONEME 6-BIT CODE TO SYNTHESIZER (OUTPUT)
14 'PORT A,PA6 AND PA7 ARE INFLECTION I1 AND I2 RESPECTIVELY(OUTPUTS)
15 POKEA+1,0:POKEA,255:POKEA+1,52
19 'ENABLE CC SOUND MUX INPUT FROM CARTRIDGE
20 POKE65281,180:POKE65283,61:POKE65315,60
28 'SEND STOP PHONEME
30 P=63:GOSUB1000
50 PRINT:PRINT"PHONEME STRING TALKER":PRINT
90 'DATA FOR SIGN-ON MESSAGE AND ABC'S.
94 DATA27,47,24,52,53,55,62,62,21,0,9,47,0,12,12,56,60,60
95 DATA23,23,21,24,38
96 DATA25,29,58,49,12,37,34,54,55,42,58
97 DATA42,61,25,58
98 DATA62,62,62
99 DATA6,33,41,14,60,41,31,60,41,30,60,41,60,41,2,1,29,38,26,60,41
100 DATA6,33,41,42,16,21,0,9,41,30,26,0,6,33,41,25,0,6,33,41,2,0,35,24,2,1,12,2,
1,13,52,53,55,37,60,41,62,23,34,34,55,55,21,49,58,2,1,31,42,00
110 DATA33,41,34, 4,35,33,15,60,33,41,30,58,14,35,24,34,54,54,2,1,23,31,31,43,21
,0,9,41,18,60,41
190 '143 PHONEMES TO OUTPUT
200 FOR N=1TO143:READZ(N):NEXTN
205 N=N-1
210 FORI=1TON:P=Z(I):GOSUB1000
220 NEXTI
298 'MAIN LOOP AND ROUTINE
299 'SILENCE TALKER,GET NEW OR REPEAT OLD PHONEME,OUTPUT TO SYNTHESIZER
300 P=63:GOSUB1000
400 GOSUB 5000
500 FORI=1TON:P=Z(I):GOSUB1000
700 NEXTI:GOTO300
999 'TALK OUTPUT ROUTINE
1000 POKEA+1,52 'STB HIGH
1010 POKEA,P 'PLACE 6-BIT PHONEME CODE AND INFLECTION ON PORT A
1020 POKEA+1,60 'STB LOW
1022 V=PEEK(A) 'DUMMY READ TO RESET CA1
1030 IF (PEEK(A+1)AND128)THENRETURN ELSE 1030 'RETURN WHEN A/R GOES HIGH
5000 'LOAD PHONEME STRING
5040 PRINT:PRINT"PRESS @ KEY TO PLAY OLD STRING"
5100 PRINT"OR INPUT NEW STRING XX,XX, ETC."
5140 AN$=""
5200 A$=INKEY$:IFA$=""THEN5200
5210 PRINTA$;
5220 AN$=AN$+A$
5230 IFA$="@"THENRETURN
5300 IFA$=CHR$(13)THEN5500ELSE5200
5500 'DECODE STRING
5510 ST=1:I=1:P$="":A$=""
5520 'P$=NEXT PHONEME WHEN DONE AND IS MADE UP USING A$
5529 'AN$=ENTIRE STATEMENT INCLUDING DELIMITERS (SPACE OR COMMAS)
5530 P$=P$+A$
5540 A$=MID$(AN$,ST,1)
5550 'ST IS COUNTER USED TO STEP THROUGH AN$
5560 ST=ST+1
5570 IF A$=CHR$(44)ORA$=CHR$(32)ORA$=CHR$(13)THENGOSUB6000ELSE5530
5600 IF A$=CHR$(13)THEN5700
5610 A$=""
5620 GOTO 5530
5700 N=I-1:RETURN
6000 'PHONEME CODES
6100 IFP$="EH3"THENP=0
6101 IFP$="EH2"THENP=1
6102 IFP$="EH1"THENP=2
6103 IFP$="PA0"THENP=3
6104 IFP$="DT"THENP=4
6105 IFP$="A2"THENP=5
6106 IFP$="A1"THENP=6
6107 IFP$="ZH"THENP=7
6108 IFP$="AH2"THENP=8
6109 IFP$="I3"THENP=9
6110 IFP$="I2"THENP=10
6111 IFP$="I1"THENP=11
6112 IFP$="M"THENP=12
6113 IFP$="N"THENP=13
6114 IFP$="B"THENP=14
6115 IFP$="V"THENP=15
6116 IFP$="CH"THENP=16
6117 IFP$="SH"THENP=17
6118 IFP$="Z"THENP=18
6119 IFP$="AW1"THENP=19
6120 IFP$="NG"THENP=20
6121 IFP$="AH1"THENP=21
6122 IFP$="OO1"THENP=22
6123 IFP$="OO"THENP=23
6124 IFP$="L"THENP=24
6125 IFP$="K"THENP=2
6126 IFP$="J"THENP=26
6127 IFP$="H"THENP=27
6128 IFP$="G"THENP=28
6129 IFP$="F"THENP=29
6130 IFP$="D"THENP=30
6131 IFP$="S"THENP=31
6132 IFP$="A"THENP=22
6133 IFP$="AY"THENP=33
6134 IFP$="Y1"THENP=34
6135 IFP$="UH3"THENP=35
6136 IFP$="AH"THENP=36
6137 IFP$="P"THENP=37
6138 IFP$="O"THENP=38
6139 IFP$="I"THENP=39
6140 IFP$="U"THENP=40
6141 IFP$="Y"THENP=41
6142 IFP$="T"THENP=42
6143 IFP$="R"THENP=43
6144 IFP$="E"THENP=44
6145 IFP$="W"THENP=45
6146 IFP$="AE"THENP=46
6147 IFP$="AE1"THENP=47
6148 IFP$="AW2"THENP=48
6149 IFP$="UH2"THENP=49
6150 IFP$="UH1"THENP=50
6151 IFP$="UH"THENP=51
6152 IFP$="O2"THENP=52
6153 IFP$="O1"THENP=53
61 4 IFP$="IU"THENP=54
6155 IFP$="U1"THENP=55
6156 IFP$="THV"THENP=56
6157 IFP$="TH"THENP=57
6158 IFP$="ER"THENP=58
6159 IFP$="EH"THENP=59
6160 IFP$="E1"THENP=60
6161 IFP$="AW"THENP=61
6162 IFP$="PA1"THENP=62
6163 IFP$="STOP"THENP=63
6190 Z(I)=P+IN
6199 'INFLECTION BITS ARE ADDED TO PHONEME CODES
6200 IFP$="IN0"THENIN=0
6210 IFP$="IN1"THENIN=64
6220 IFP$="IN2"THENIN=128
6230 IFP$="IN3"THENIN=192
6240 I=I+1
6250 P$=""
6300 RETURN
```

HI-RES COLOR GRAPHICS

Thomas H. Hunt
30001 Wagner
Warren, MI 48093

How would you like to latch onto a color graphics chip that could easily simulate realistic three-dimensional images, provide real time animation with minimal programming effort, plus a high resolution (256 x 192) bit-mapped plane -- all in fifteen distinct colors? Or what about hooking this chip up to your video tape recorder and have your computer provide subtitling and animated graphics overlays to any video tape? Or adding a 24 x 40 text display, with a completely programmable 256-character set,to boot? If this interests you, allow me to introduce you to the TEXAS INSTRUMENTS TMS-9918A Video Display Processor (VDP).

Perhaps I should say "reintroduce", for TI first announced this version of the chip about mid-1979. A couple of years prior to that, the first version of the chip (without the "A" suffix) was produced. It fell into almost instant obscurity,never to attain the level

of publicity and availability that chips like the MC-6847 received. In an attempt to turn things around, TI added a full graphics mode and announced the "A" version. It became readily available in late 1980 and found its way into the TI 9900 home computer, an S-100 board, and an add-on board for PET/TRS-80 computers.

THE NIGHT THE SPRITES WENT ON IN GEORGIA

Just recently, however, the TMS-9918A VDP has finally been put to some really meaningful use -- it is now available on the SS-50 buss! TERMINUS DESIGN in Ellenwood, Ga. has seen fit to produce a board called the ARCADE-50. This board is based around the TMS-9918A,plus three GI sound generators, an A/D converter, and a parallel I/O port. In short, it has everything necessary to create a spectacular animated display with an SS-50 computer.

Understanding the ARCADE-50 really boils down to being able to understand and create a display with the TMS-9918A. Simply writing a cursory review of the ARCADE-50 would be doing this fine effort an injustice, thus Part I will concentrate on what can be accomplished with the TMS-9918A.

A 40-page Technical Manual is yours for the asking at most TI Sales Offices, or it is included with purchase of the ARCADE-50. This is almost required reading. It is a bit disorganized, but otherwise well written, and all the gory technicalities are there. To reinforce with a different point of view, I will try to leave you with an overall feel for the visual effects that can be created. Once the conceptual idea is firmly implanted, the actual mechanical details can easily be gleaned from the TI manual. I will also discuss some not-so-obvious techniques for creating some very interesting effects.

DISPLAY CONCEPTS

The technique used by the TMS-9918A in creating the display was first pioneered by none other than Walt Disney. It can perhaps best be envisioned by picturing a deck of 36 large, transparent cards stacked one on top of the other. Images are then painted on each card (image plane). At this point, each image plane can be independently moved around, in its own X, Y direction, creating motion relative to the other planes. The plane's Z-axis (depth) in the deck of cards, of course, remains constant. When appropriately drawn, the result is a very effective composite scene, reminiscent of a stage play or cartoon.

When images created on one plane interfere with images on another, a priority system resolves which one is actually displayed. The first plane, or foreground,has the highest priority, and any image painted on it will automatically mask an image behind it. The second plane has the next highest priority, etc.

ADD A LITTLE SPRITES TO YOUR LIFE

The first 32 planes are actually special object-oriented planes called "Sprites". They are limited in scope -- one color per sprite and either 8 x 8, 16 x 16, or 32 x 32 pixels per sprite -- but they are highly mobile. They can be positioned anywhere on the screen by simply storing the desired X and Y coordinates in a pair of VDP registers. Outside of the pixel pattern, the rest of the sprite plane is transparent.

With 32 sprite planes available simultaneously, a tremendous amount of action can be portrayed on the screen. Although each sprite is limited to one color, larger, multicolored objects can be created by concantinating several sprites. Incidently, this is the same concept used by ATARI for their "player-missile" graphics.

NOW ADD A FULL PATTERN PLANE

While the sprites were intended to create the highly mobile actors or objects, no scene would be complete without some form of background. Indeed, the plane immediately behind the sprite planes accomplishes this task. Called the Pattern Plane, it can be thought of as the traditional bit-mapped graphics plane (although this is not its only application).

Unlike the sprites, which are limited to a small number of pixels, the pattern plane has the full capability of 256 x 192 pixels. It can be a solid color, a mixture of colors, a mixture of solid and transparent, or all transparent rendering it invisible. As it is behind the sprite planes, it has lower priority. Thus any pattern on a sprite will automatically mask whatever image is placed on the pattern plane. It might be interesting to reflect for a moment and note that the entire graphics capability of machines like the Apple and the Color Computer consist solely of a "pattern plane".

A LOOK BEHIND THE SCENES

After the pattern plane appears another, larger one called the Backdrop Plane. It appears as a solid color and serves to form a border around the other video display elements. It can be set instantly to any of the fifteen colors, or transparent, by updating one control register in the VDP.

Behind the backdrop, provisions have been made for a very interesting plane -- the External Video Input. This allows the user to input a standard broadcast signal, perhaps from a video tape recorder (VTR) or a video camera. By setting the backdrop to transparent, text and graphics can be overlayed on the input signal. The mixed output can then be displayed on a color TV monitor or recorded on another VTR. This would be a very useful feature for experimenting with subtitling or interactive live and animation broadcasting.

The final plane in the series, and the one with the least priority, encompasses the entire screen. It is permanently all black, i.e. when all colors in the system are programmed to be transparent, and external video is turned off, the display will appear blank.

STREET SCENE

It takes very little imagination to realize the new realms of animation possibilities opened up by the multiple plane concept. For example, picture a stationary street scene drawn on the pattern plane, and a car made with sprites, driving across the screen. Buildings and pedestrians in the background can be seen through the car's windows as it passes. Now a lamppost or tree is drawn with higher priority sprites on the foreground side of the street. The car will smoothly pass behind these objects -- all with minimal programming effort!

A HARD WARE'S KNIGHT

To create the display, the VDP utilizes up to 16K of RAM to store the video information. Fortunately, this 16K of video RAM (VRAM) is not directly connected to the system buss. Rather, the VDP takes complete responsibility for management of the VRAM. To the CPU, the entire operation appears strictly as a parallel port.

Therefore, the VDP is not really memory-mapped in the sense that the CPU does not directly access the VRAM. All CPU accesses to the VRAM are pipelined through the VDP. Although this method is somewhat slower than direct CPU access, the other features of the VDP tend to considerably lessen this impact.

There are a couple of big advantages to an I/O port concept: 1) It doesn't steal 16K of address space from YOUR memory map, and 2) as the VDP handles all memory accesses, there is absolutely no screen flicker during memory updates. The CPU must intervene only to load the VDP with the proper data and to lead the inevitable initialization registers.

PLAY IT AGAIN, VRAM

The next hurdle to get over is understanding how the data is organized inside VRAM to achieve a desired display. First, bear in mind that data in VRAM affects only the 32 sprite planes and the pattern plane. The backdrop plane is a solid color which can be changed by updating a VDP register, and the external video plane, of course, requires an external broadcast signal.

The VDP departs from the linear bit-by-bit mapping of memory onto raster lines of the screen. Instead a more complex, but much more flexible, scheme is used. The VRAM is actually divided up into a series of tables that contain pointers to each other as well as pointers to a screen area. The data for each table is prepared by the programmer and then loaded into VRAM. Within certain limitations, the tables may be placed anywhere in VRAM. VDP registers are then given the starting address of each table.

TABLE FOR TWO

Two tables must be defined in VRAM in order to create Sprites -- the Sprite Attribute Table (SAT) and the Sprite Generator Table (SGT). The SAT specifies sprite color, its position on the screen, and a pointer to the desired pattern. The SGT contains the actual bit patterns that describe what the sprite will look like. A queue of up to 256 different patterns may be predefined in the SGT. Pointers in the SAT then pick as many as 32 of the sprites to be simultaneously displayed.

Sprites are available in four sizes:

Actual Pixels	Mag.	Effective Pixels
8 x 8	x 1	8 x 8
8 x 8	x 2	16 x 16
16 x 16	x 1	16 x 16
16 x 16	x 2	32 x 32

Unfortunately, this selection affects all sprites. The only other limitation to sprites is that only four may appear on any one raster line at a time. The fifth, or more, offending sprites will simply not be displayed on that particular raster line.

EASY MOVIN'

The problem of maintaining the integrity of two-dimensional objects with traditional one-dimensional linear displays has always been difficult to resolve. In these displays, the object appears two dimensional on the screen,but the data that makes up the object is actually scattered throughout the video memory. Even simple movement requires an ungodly amount of software overhead. The problem of moving one object behind another requires even more taxing hidden line algorithms.

Sprite planes provide an elegant solution to these problems. Movement is accomplished by simply updating a pair of X,Y pointers in the SAT. Passing behing or in front of another object is handled in hardware -- being on different planes, they merely seem to interact. Also, if the programmer finds it necessary to dynamically change the sprite pattern, he will find the data neatly grouped in a contiguous area of the VRAM.

In addition to rapid, easy movement,several other interesting effects can be created with sprites. A sprite can be accented by rapidly cycling its color through the 15 combinations -- or even made to disappear by making it transparent. In place rotation can be simulated by defining the rotational iterations in several sprites. By rapidly changing the pattern pointer in the SAT, the various patterns will be displayed at that location, giving the illusion of rotation. Other internal movement can also be obtained using this technique (PAC-MAN chomping?).

PATTERN PLANE STRUCTURE

Setup and programming of the Pattern Plane is the most involved part of using the TMS9918A. The Pattern Plane is activated by defining three additional tables in VRAM -- the Pattern Name Table, the Pattern Generator Table, and the Pattern Color Table. To further confuse matters, the Pattern Plane can also be placed in one of four modes. The selected mode tells the VDP how to interpret the data found in VRAM (in terms of color, pattern, and resolution). Each mode, therefore, will use the tables in a slightly different fashion.

Images on the pattern plane are created in a character oriented format. Picture the screen as a 32 x 24 grid numbered sequentially from 0 to 767, top left to lower right. This is the visual interpretation of the Pattern Name Table. Each entry in this 768 byte table corresponds to one of the 768 squares on the screen. However, instead of containing the actual data to be displayed, the entries form an index into both the Pattern Generator Table and the Pattern Color Table.

The Pattern Generator Table contains the actual pixel pattern that will be displayed on the screen -- 8 bytes or 64 pixels per square. Similarly, the color of a given pattern is determined by individual entries in the Pattern Color Table. This is exactly equivalent to the screen memory on all CRT terminals. Data in screen memory is actually just an index into an ASCII character generator.

This three table system for displaying an image offers the programmer a tremendous amount of flexibility. The programmer can arrange the data to represent alphanumeric characters, special characters, bit-mapped pixels, or combinations thereof. It is quite easy to construct a display consisting of special graphics characters, text, and hi-res pixels simultaneously. In addition, the shape, color, and position of the patterns can all be separately manipulated by updating the individual table entries.

IN THE MODE

The Pattern Plane can be placed in one of four graphics modes. Graphics Mode I and Graphics Mode II work on the three table principle just described. These modes are virtually identical except that GM-I has a much more limited Pattern Generator Table (2K long) and Color Table (32 bytes). In GM-II, both of these tables are 6K bytes long. This is sufficient to set up a bit-mapped plane with 256 x 192 pixel resolution and 32 x 192 Color resolution.

The Multicolor Mode is a coarse graphics mode of 64 x 48 resolution. Each block can only be on or off, but may be any one of the fifteen available colors. And finally, Text Mode is a special case of GM-I. The grid was expanded to allow a 24 x 40 character display. Only two colors are allowed in this mode, but they can be any two of the fifteen colors. There is no internal character generator, so the user is free to load the Pattern Generator Table with any desired character set. Also, sprites are unavailable in the Text Mode.

A WALK ON THE WILD SIDE

The uses and advantages of Graphics Mode II and Sprites are fairly obvious and have already been discussed. Graphics Mode I, however, has a couple of interesting possibilities that are not so readily obvious. On the surface, it would appear that its only use would be in a programmable character generator - type environment. Nothing wrong with this -- a considerable amount of animation can be achieved using this alone.

Note, however, that GM-I requires only 2K bytes for a pattern generator table while the VDG supports 16K. This means that there is room in VRAM to predefine multiple screens -- seven different screens to be exact! These screens can be quickly called up by simply updating one VDG register. By cleverly designing each screen and then rapidly cycling through them an amazing degree of animation can be accomplished, especially with Sprite enhancements. This technique, called multiple-buffering, can also be used with Multicolor and Text Modes.

The limited number of color entries (32) in the GM-I Pattern Color Table suggests another possibility. In certain patterns rapid motion can be simulated, not by changing the pattern, but by changing the color arrangement of the pattern! Every reader has probably seen the apparent movement down the trench in a well known home arcade game!

In summary, the TMS-9918A offers an elegant solution to object integrity under rapid motion (via the Sprite feature). It also combines the best features of a programmable character generator and a bit - mapped graphics system. It provides for multiple frame buffering as well as independent control over an object's color, position, and shape.

LIMITATIONS

Man has yet to create anything so perfect that it pleases everyone. And the TMS9918A is certainly no exception. To be realistic, remarks in this section will be confined to features that are missing on the TMS9918A but are included on other existing chips.

It is surprising to note that, with the heavy emphasis placed on Sprites, TI only implemented the most rudimentary form of Sprite collision detection. One bit,buried in a VDP register, informs the programmer that somewhere on the screen at least two Sprites have overlapping "on-bits". There is no provision for Sprite to Pattern Plane coincidence checking.

Being able to quickly and accurately pinpoint collisions is tantamount to the success of any arcade game application. The programmer is now given the added burden of maintaining and continually cross checking a complete set of absolute position coordinates. In contrast, the ATARI computer provides 60 bits of collision status information!

Horizontal and vertical fine scrolling is another stupendous effect that cannot be effectively created with the TMS9918A. Thus one cannot easily create a large scene in memory and then use the screen as a "window" to pan over this scene. In the TMS-9918A such effects must be created by brute force mass memory move or sprite-concantination techniques.

It would have been far better to go one level deeper with the Color Table. Instead of having each entry in the table point to a absolute screen position, it should point to one of several color indirection registers. The contents of the color indirection registers would then determine the actual displayed color. This would not only provide a fast means for mass color changes, but also allow up to 256 hue/intensity combinations. Color indirection would add as much power over other chips in its class as indirect addressing did in the 6809.

DIM SCREEN, FADE OUT...

All in all, the TMS-9918A solves many more graphics problems than its few limitations create. After working with this chip for several months, I am still amazed at the effects that can be created. The next installment in this series will take a closer look at the ARCADE-50 board and the FBASIC software -- a compiled BASIC, no less,with emphasis on easy creation of graphics programs.

SDS80C - FLEX

ADAPTING THE MICROWORKS SDS80C TO FLEX09 DOS

I've been using the Microworks SDS80C Software Development System for about a year on my TRS80C and have been very pleased with it except for one problem--the very unreliable tape load system of the TRS80C. It's very disconcerting to work for hours on a program only to lose it because it can't be reloaded from tape. (Radio Shack should have a long talk with JPC Prods. about their TC-3 system.) This was my main reason for going to FLEX and the FMATE(RS) system. Neat as FLEX is in general, their Editor/Assembler is much clumsier for me to use than the very friendly SDS80C. Besides, I have a lot of programs on tape that I wanted into the FLEX FMS; therefore, I decided to try to adapt SDS80C to FLEX.

What follows is a step-by-step procedure that will accomplish the adaptation. Be advised that my version of the related programs are FLEX V3.01, SDS80C Rev 1.3 and, of course, BASIC 1.1. What happens with other versions of these programs is something I can't predict.

Step 1: Save SDS on tape.

A) Execute SDS80C.
B) Using ABUG "T", move SDS from C000-DFFF to 4000-5FFF.
C) Using ABUG "M", change $4016 to "BD A1C1" (changes a LBSR to JSR)
D) Using ABUG "S", save on tape as "4000 5FFF 4000 SDS".

Step 2: Move RS BASIC to FLEX.

A) Move Basic BASIC to FLEX as described by FMAT(RS) SAVEROM.CMD.
B) APPEND RSBASIC with either the listing described in SAVEROM or the one given here only through line 68 deleting lines 18, 19, 20, 35, 36.

Step 3: Append SDS and RSBASIC.

A) Execute RSBASIC to get Radio Shack BASIC.
B) Load from tape and execute SDS in low memory. CLOADM:EXEC.
C) Using ABUG "T", move SDS from 4000-5FFF to 8000.
D) Go back to FLEX. ABUG "J CD03".

Step 4: Modify SDS80C+BASIC.

A) SAVE SDSXXX.BIN 8000 BFFF A027
B) Using FLEX Editor/Assembler, write and assemble SDSFLEX per the listing.
C) APPEND SDSXX.BIN SDSFLEX.BIN SDSZZZ.CMD
D) Execute SDSZZZ. This should bring up the familiar "EDITOR" banner.
E) Return to FLEX. ABUG "J CD03".
F) Make a clean copy on another disk by SAVE SDS80C.CMD 8000 BFFF A027.

You should now have SDS80C as a FLEX command. Loading text from tape is still R-L or R-A (most of the time!). Access to FLEX is via ABUG and is pretty much described by the listing. The commandsG,R,U and ! are lost or reassigned. "K" now "keeps" machine language stuff on tape and "E" returns to Editor from ABUG. "F" calls FLEX as a subroutine for CAT, SAVE, etc. "G" gets, "A" appends, and "S" saves SDS Editor text files; "S" finds the address of the text itself--just input file name. Use FLEX to SAVE and GET binary files. J CD03 will return to FLEX and MON (in FLEX) goes back to the SDS Editor with the text intact.

The printer routine has been modified to run at 2400 Baud and to provide a form feed and header after 60 lines. The header includes the program NAM and page numbering (after page 1). Start the paper about two lines down from the perforation.

The excellent SIGMON program has been similarly adapted to FLEX. The steps are much the same as those described here with the same program listed through line 68 deleting lines 18, 19, 32, 33, 46-53.

STEWART D. LYON W6CUX
19943 ARMINTA ST.
WINNETKA, CA 91306

Editor's Note: The above program from Stewart Lyon was accompanied by a letter. In his letter was a very interesting remark, "The software of the recent Conestoga I Rocket launched in Texas was developed on the very same TRS80C I'm typing this letter on!"

For those who keep trying to tell me that the Color Computer is a 'toy', well.

The Color Computer is an excellent vehicle to get started with. It can, and does, exceed the production of any other 'home' type computer, especially when outfitted with FLEX™. Stewart runs the Data-Comp F-Mate(RS) FLEX™ disk system with his Color Computers. I know of many who started with the RS Color Computer and FLEX™, and later upgraded to fullbore Standard S50 Bus 6809 computers.

I have written Stewart and asked him to give us all some background on the Conestoga I project. Should be very interesting reading. Will publish it, if and when we receive it from Stewart.

DMW - - -

```
SDSFLEX                          10-23-82 TSC ASSEMBLER   PAGE    1

                          ***************************
                          * ADAPT MICROWORKS SDS80C
                          * TO FLEX09 FOR TRS80C.
                          * RETAINS TAPE COMMANDS AND
                          * ADDS ACCESS TO FLEX FMS
                          ***************************

                          * S.D.LYON
                          * 19943 ARMINTA ST.
                          * WINNETKA, CA 91306

A017                             ORG    $A017
A017 4F                          CLRA                 TURN OFF DISK DRIVE
A018 B7    FF40                  STA    $FF40
A01B 8E    A089                  LDX    #LINK         LINK BACK TO EDITOR
A01E BF    CC9D                  STX    $CC9D         IN CASE OF DISK ERROR
A021 20    51                    BRA    $A074         USE "MON" TO RETURN

A055                             ORG    $A055
A055 0D                          FCB    $0D           LIMIT RANGE OF SAM RESET

A070                             ORG    $A070
A070 1212                        FDB    $1212         DON'T RESET SAM MEM SIZE

           9FA0   EDWARM         EQU    $9FA0
A084                             ORG    $A084
A084 8E    7FF0                  LDX    #$7FF0        STACK AT TOP OF LOWER RAM
A087 20    0A                    BRA    $A093
A089 BD    B9DB   LINK           JSR    SDSF3
A08C 7E    9FA0                  JMP    EDWARM

A0CB                             ORG    $A0CB
A0CB 7E    8000                  JMP    $8000         JUMP TO SD580C AFTER SETUP

A0FC                             OR     $A0FC
A0FC 3B                          RTI                  DON'T DO ROM PAC

A113                             ORG    $A113
A113 0012                        FDB    $12           PRINTER AT 2400 BAUD

                          * ALTER PRINTER ROUTINE TO DO
                          * FORM FEED AFTER 60 LINES
A2DD                             ORG    $A2DD
A2DD 27    16                    BEQ    $A2F5
A2E3                             ORG    $A2E3
A2E3 C1    0                     CMPB   $00
A2F5                             ORG    $A2F5
A2F5 BD    BC94                  JSR    LFCTR
A2F8 20    ED                    BRA    $A2E7
A2FA 12                          NOP

A1A5                             ORG    $A1A5
A1A5 84    60                    ANDA   #$60          BLINKING CURSOR
A1A7 88    40                    EORA   #$40

AADE                             ORG    $AADE         "BASIC" LINK TO "FLEX"
AADE 46 4C 45 58                 FCC    'FLEX'        NOT USED IN THIS PATCH
AAE2 80                          FCB    $80

ABEE                             ORG    $ABEE
ABEE 3E3E                        FDB    $3E3E         BASIC "OK" TO ">>"-NOT USED

A7BD                             ORG    $A7BD
A7BD 1A    50                    ORCC   #$50          BASIC TO FLEX WARMS-NOT USED
A7BF 7E    CD03                  JMP    $CD03

B9B9                             ORG    $B9B9         SDS-FLEX LINK
B9B9 BD    A1B1   GOFLX          JSR    $A1B1         WAIT FOR KEY PRESS
B9BC 86    03                    LDA    #3            FLEX USES DIRECT PAGE
B9BE 1F    8B                    TFR    A,DP          AT RS TEXT BUFFER
B9C0 BD    CD24                  JSR    $CD24         FLEX CRLF
B9C3 BD    CD1B                  JSR    $CD1B         FLEX INBUF
B9C6 BD    CD4B                  JSR    $CD4B         FLEX DOCMND
B9C9 4F                          CLRA                 TURN OFF DISK DRIVE
B9CA 1F    8B                    TFR    A,DP          RESTORE DIRECT PAGE
B9CC B7    FF40                  STA    $FF40
B9CF BD    A1B1                  JSR    $A1B1         WAIT FOR KEY
B9D2 20    12                    BRA    SDSF2

B9D4 86    03     SDSF1          LDA    #3
B9D6 1F    8B                    TFR    A,DP
B9D8 BD    C 4B                  JSR    $CD4B         FLEX DOS AS SUBRT
B9DB 4F           SDSF3          CLRA                 TURN OFF DISK DRIVE
B9DC 1F    8B                    TFR    A,DP          RESTORE DIRECT PAGE
B9DE B7    FF40                  STA    $FF40
B9E1 86    4F                    LDA    #$4F          RESTORE CODE
B9E3 B7    D21E                  STA    $D21E         IN FLEX "GET"

B9E6 8E    0618   SDSF2          LDX    #$0618
B9E9 9F    00                    STX    0             RESTORE START OF TEXT PTR

B9EB 8E    0400                  LDX    #$0400        RESTORE SCREEN MEMORY LOC
B9EE 34    10                    PSHS   X
B9F0 8E    FFC6                  LDX    #$FFC6
B9F3 86    06                    LDA    #6
B9F5 1F    89     DISP3          TFR    A,B
B9F7 68    61                    ASL    1,S
B9F9 69    E4                    ROL    0,S
B9FB 59                          ROLB
B9FC A7    85                    STA    B,X
B9FE 4A                          DECA
B9FF 2A    F4                    BPL    DISP3
BA01 35    96                    PULS   D,PC

                          * APPEND TEXT FILES
BA03 DC    02     APPEND         LDD    2             END OF CURRENT TEXT
BA05 83    0600                  SUBD   #$0600        REMOVE PAGE OFFSET -5
BA08 FD    CC1B                  STD    $CC1B         OFFSET LOADER LOC
BA0B 86    39                    LDA    #$39          CHANGE CODE IN FLEX
BA0D B7    D21E                  STA    $D21E         A DEFINITE NO-NO
BA10 31    8D 0266               LEAY   APPMS,PCR
BA14 A6    A0     APP1           LDA    ,Y+
BA16 27    05                    BEQ    APP2
BA18 BD    A30A                  JSR    $A30A         PRINT ON SCREEN
BA1B 20    F7                    BRA    APP1
BA1D 8E    020E   APP2           LDX    #$020E
BA20 A6    A0     APP3           LDA    ,Y+           PUT "GET" IN BUFFER
BA22 27    0A                    BEQ    GET1
BA24 A7    80                    STA    ,X+
BA26 20    F8                    BRA    APP3

                          * GET TEXT FILES
BA28 31    8D 0259 GET           LEAY   GETMS,PCR
BA2C 8D    20                    BSR    CMD           PUT CMD IN RS BUFFER
BA2E 8D    2C     GET1           BSR    CMD2
BA30 8D    34                    BSR    XFER          TRANSFER TO FLEX BUFFER
BA32 8D    A0                    BSR    SDSF1         CALL FLEX
BA34 BE    CC3D                  LDX    $CC3D         END OF FILE
BA37 9F    02                    STX    2             TO SDS END OF TEXT
BA39 7E    9FA0                  JMP    EDWARM        RETURN TO EDITOR

                          * SAVE TEXT FILES
BA3C 31    8D 024C SAVE          LEAY   SVEMS,PCR
BA40 8D    0C                    BSR    CMD
BA42 8D    18                    BSR    CMD2
BA44 8D    37                    BSR    ADDR          TEXT ADDR RANGE TO BUFFER
BA46 8D    1E                    BSR    XFER
BA48 17    FF89                  LBSR   SDSF1
BA4B 7E    9FA0                  JMP    EDWARM
                          * PUT COMMAND IN RS BUFFER
BA4E 8E    020E   CMD            LDX    #$020E        RS LINE BUFFER
BA51 A6    A0     CMD1           LDA    ,Y+
BA53 27    10                    BEQ    CMD3
BA55 A7    80                    STA    ,X+
BA57 BD    A30A                  JSR    $A30A
BA5A 20    F5                    BRA    CMD1
BA5C 34    10     CMD2           PSHS   X
BA5E 5F                          CLRB
BA5F BD    A39A                  JSR    $A39A         FILE NAME IN RS BUFFER
BA62 35    10                    PULS   X
BA64 3A                          ABX                  POINT TO NEXT LOC IN BUFFER
BA65 39           CMD3           RTS

                          * TRANSFER FROM RS BUFF TO FLEX BUFF
BA66 86    0D     XFER           LDA    #$0D
BA68 A7    84                    STA    ,X
BA6A 8E    C080                  LDX    #$C080        POINT TO FLEX BUFFER
BA6D BF    CC14                  STX    $CC14         SAVE POINTER
BA70 108E  02E0                  LDY    #$2E0         POINT TO RS BUFFER+2
BA74 A6    A0     XFER1          LDA    ,Y+           COPY UNTIL AFTER CR
BA76 A7    80                    STA    ,X+
BA78 81    0D                    CMPA   #$D
BA7A 26    F8                    BNE    XFER1
BA7C 39                          RTS

                          * WRITE START & END ADDRESS
                          * OF TEXT INTO RS BUFFER
BA7D 86    20     ADDR           LDA    #$20
BA7F A7    80                    STA    ,X+
BA81 108E  0000                  LDY    0
BA85 8D    04                    BSR    ADDO
BA87 86    20                    LDA    #$20
BA89 A7    80                    STA    ,X+
BA8B 8D    00     ADDO           BSR    OUT2H
BA8D A6    A4     OUT2H          LDA    ,Y
BA8F 8D    04                    BSR    OUTHL
BA91 A6    A0                    LDA    ,Y+
BA93 20    04                    BRA    OUTHR
BA95 44           OUTHL          LSRA
BA96 44                          LSRA
BA97 44                          LSRA
BA98 44                          LSRA
BA99 84    0F     OUTHR          ANDA   #$F
BA9B 8B    30                    ADDA   #$30
BA9D 81    39                    CMPA   #$39
BA9F 23    02                    BLS    OUTCH
BAA1 8B    07                    ADDA   #7
BAA3 A7    80     OUTCH          STA    ,X+
BAA5 39                          RTS

BC7A                             ORG    $BC7A
BC7A 08 08        APPMS          FCB    8,8
BC7C 41 50 50 45                 FCC    /APPEND /
BC84 00                          FCB    0
BC85 08 08        GETMS          FCB    8,8
BC87 47 45 54 20                 FCC    /GET /
BC8B 00                          FCB    0
BC8C 08 08        SVEMS          FCB    8,8
BC8E 53 41 56 45                 FCC    /SAVE /
BC93 00                          FCB    0

                          * FORM FEED AFTER PRINTING 60 LINES
BC94 C6    3B     LFCTR          LDB    #59
BC96 F1    BCF5                  CMPB   LINE
BC99 22    09                    BHI    LFCT1
BC9B 7F    BCF5                  CLR    LINE
BC9E 86    0C                    LDA    #$C           FORM FEED
BCA0 8D    06                    BSR    PRINT         ADVANCE ACROSS PERF.
BCA2 20    07                    BRA    PPAGE         PRINT HEADER
BCA4 7C    BCF5   LFCT1          INC    LINE
BCA7 39                          RTS

BCA8 7E    A2BF   PRINT          JMP    $A2BF

                          * PRINT NAME AND PAGE NUMBER
BCAB 34    36     PPAGE          PSHS   D,X,Y
BCAD 7C    BCF6                  INC    PAGE          PAGE COUNT
BCB0 8E    01D2                  LDX    #$1D2
BCB3 8D    2D                    BSR    PTXT          PRINT NAME
BCB5 C6    3A                    LDB    #58
BCB7 8D    21                    BSR    PSPC
BCB9 4F                          CLRA
BCBA F6    BCF6                  LDB    PAGE
BCBD 31    8D 002E               LEAY   NUM,PCR       FORMAT PAGE NUMBER
BCC1 8E    00F9                  LDX    #$00F9        POINT TO 100'S
BCC4 BD    000F                  JSR    $000F         SDS HEX/DEC ROUNTINE
BCC7 C6    03                    LDB    #3
BCC9 A6    A2     DEC1           LDA    ,-Y
BCCB 84    3F                    ANDA   #$3F          MASK "$60" AS A SPACE
BCCD A7    A4                    STA    ,Y
BCCF 5A                          DECB
BCD0 26    F7                    BNE    DEC1
BCD2 30    8D 0014               LEAX   PGMS,PCR
BCD6 8D    0A                    BSR    TXT           PRINT "PAGE----"
BCD8 35    B6                    PULS   D,X,Y,PC      RETURN
                0
```

```
246  BCDA B6  20       PSPC    LDA     #$20        PRINT SPACES
247  BCDC BD  CA       PSP1    BSR     PRINT
248  BCDE 5A                   DECB
249  BCDF 26  FB               BNE     PSP1
250  BCE1 39           PSP2    RTS
251
252  BCE2 A6  80       PTXT    LDA     ,X+         PRINT TEXT AT X
253  BCE4 27  FB               BEQ     PSP2
254  BCE6 BD  C0               BSR     PRINT
255  BCE9 20  F8               BRA     PTXT
256
257  BCEA 50 41 47 45  PGMS    FCC     /PAGE /
258  BCEF 20 20 20     NUM     FCC     /   /
259  BCF2 00 00 00             FCB     $0,$0,$0
260
261  BCF5 00           LINE    FCB     0
262  BCF6 00           PAGE    FCB     0
263                    * DETOUR TO CLEAR LINE AND PAGE COUNTER
264  BCF7 7F  BCF6     CLRLC   CLR     PAGE        PAGE COUNTED
265  BCFA 7C  BCF6             INC     PAGE
266  BCFD 7F  BCF5             CLR     LINE        LINE COUNTED
267  BD00 7E  0237             JMP     $237
268
269                    * PATCHES TO SDGBBC EDITOR AND ASUG
270  B816                      ORG     $B816
271  B816 BD  A1C1             JSR     $A1C1       LBSR TO JSR
272  B305                      ORG     $B305
273  B305 BD  BCF7             JSR     CLRLC
274
275                    * PATCHES TO ASUG
276                    * REASSIGNS KEYS
277  9FBC                      ORG     $9FBC
278  9FBC 41                   FCC     'A'
```

```
SDGFLEX                          10-23-82 TSC ASSEMBLER   PAGE    6

279  9FBD 1A46                 FDB     APPEND-1    APPEND TEXT
280  9FBF 47                   FCC     'G'         = GET TEXT FROM FLEX FILE
281  9FC0 1A4B                 FDB     GET-1
282  9FC2                      ORG     $9FC2
283  9FC2 53                   FCC     'S'         = SAVE TEXT IN FLEX FILE
284  9FC3 1A78                 FDB     SAVE-1
285  9FC5 4B                   FCC     'K'         =KEEP M.L. ON TAPE (WAS 'S')
286  9FD7                      ORG     $9FD7
287  9FD7 45                   FCC     'E'         = RETURN TO EDITOR (WAS 'B')
288  9FDA                      ORG     $9FDA
289  9FDA 46                   FCC     'F'         = CALL FLEX AS A SUBRT
290  9FDB 19DE                 FDB     DOFLX-1
291                             END

0 ERROR(S) DETECTED
```

SPELLB review

SPELLB, A Spelling Checker

SPELLB is a spelling checker from Palm Beach Software in Lantana Florida. I received a copy of a preliminary version several weeks ago, and have had extensive correspondence with Palm Beach Software, which resulted in some major improvements in the operation and execution time. They have coded their dictionary very efficiently, with codes for many normal endings for words. They claim that their compressed dictionary uses on the average, 1.7 bytes per word!

I tested the system with a 93 sector text file that was a chapter in my book on Pascal. The chapter name (filename) is SORT.TXT. To invoke SPELLB, you put a disk in the system drive, containing SPELLB.CMD, and the dictionary files, and type SPELLB SORT. SPELLB first processes the text file, reading it and sorting the words into memory. It also checks the words against a "common words" list and eliminates many common ones before sorting, thus speeding up the sort. It took 1 minute and 54 seconds to process the text file. Next, the dictionary files are read and processed. That took 1 minute and 50 seconds. At that point, all the suspect words are output to the terminal. SPELLB found 18 suspects and reported that the file contained 3905 words.

Of the words found, three were text processing commands. I have rather mixed feelings about a spelling checker eliminating all these, because I once found a typographical error in a file, having spelled 'If' backwards so it came out 'fI'. If the checker had eliminated processor commands, the error would not have been caught, because 'fi' preceded by a period for TSC's text processor, or a comma for Stylograph, is the processor command to turn the fill mode on. I'd rather be on the safe side and look at the "fI" to see if it is a processor command or a misspelled "if". I can conceive of other errors such as an extra space in a word, that could make processor commands out of word fragments.

Of the remaining 16 words, 8 were program and procedure names I had used in describing some programs in the text.

Of the remaining 8 words, 5 were proper names. That leaves a mere 3 remaining. They were "yields", "originators", and "implementations". The last is a computer oriented word, and could be classified as a special word. Two other spelling checkers found "yields" and "originators" as well.

NOTE: I received a reply to this review from Palm Beach, and they brought up a very valid point. The words implementations, originators, and yields, are all in portions of the dictionary that have not been through a final edit. No correctly spelled words at all were found among the "letters" that had already been through the final edit and compacting process.

At this point, there is a list of suspects in memory. It includes all the words that were not found in the "big dictionary". Now the list is checked against your "special" or "personal" word file, and any words found in that file is eliminated. Your personal word list might contain computer terms, medical terms, legal words, etc. Next, the list of suspects is output to the terminal, and you have a choice of dumping the list to your printer or continuing to process it. You are shown each word in the list in turn, and have the following choices as each one is presented:

A = Add the word to the special word list.

D = Delete the word from the suspect list.

H = Help. This command causes the dictionary to be searched looking for similar words. You will be prompted 'MORE', and if you keep answering 'Y', you will eventually get a list of all the words starting with the same letter.

M = Mark the word in the text file.

R = Replace the word. (You are prompted for the correct spelling.)

V = View the word in context (later).

Now, you have the option of reviewing the list again and changing your mind with regard to these actions, before any final action is taken. If you choose to continue and update the file, the words will be handled as you have indicated. If you have asked to view any words in context, they will be shown to you in a rather nice format. They are displayed on a line by themselves, between two other lines containing the previous and the following text. At that point, you may choose to leave the word alone or change it.

The whole processing of the list is done very nicely. Some words, such as proper names are obviously not misspelled, and you can simply delete them from the suspect list. You have a choice of processing the remaining words by looking at them in context or marking them for a pass with an editor.

The "big dictionary" consists of 26 dictionary files named DICA.BIN through DICZ.BIN. If you have eight inch disk drives, or some of the newer double sided double density 5 inch drives, you can have the dictionary files all on one disk. Otherwise, they may be placed on two or more. SPELLB will prompt for more dictionary files and allow you to change disks if necessary. Palm Beach recommends putting all the dictionary files on one disk along with FLEX and your most used utilities if you have

disk capacity for 1000 sectors or more, otherwise splitting the dictionary files on two or more disks.

I should say that the times reported here were with a 1 MHZ system, using 8" drives, and both the text file and the dictionary were in double density recorded files, which speeds up the read time considerably. Operation would be considerably speeded up on a 2 MHZ system, since processing time for each dictionary file is considerable with a large text file like the one I used to test this. I must also report that further communication from Palm Beach indicates that they have already made some considerable further improvements in the text file handling and dictionary scan times. I have been promised updates, and I will report on any further performance improvements.

I don't want to get into comparisons here, and I have to be careful in doing so, because I have one spelling checker that I have customized by adding many words from my text files (which are considerable). However, I will say that this checker has the largest dictionary of the four checkers that I have to test, and that it found fewer "real words" than any of the others. It is slower than some with shorter dictionaries, but it is worth a couple of extra minutes to me to save having to look up several words in the (hardcopy) dictionary to see if they are spelled correctly. This is particularly true because once started, the process proceeds through the point where it displays the suspect word list, without further operator input. That means I can take a coffee break, go stretch, check the mailbox, etc. Also, I probably spent several hours writing a text file as large as the one used in these tests. I have found that no amount of proof reading will find all the errors in a text. The eye sees what it expects.

Overall, I would have to give this one an excellent rating.

Reviewed by:

Ron Anderson - - -

VIRTUAL MEMORY*

MORE THAN FOUR DRIVES IN FLEX?
PLUS VIRTUAL DISK DRIVERS!

Matt Scudiere (615) 482-1355 home evenings
Union Carbide (615) 576-7684 9-5 EDT work
P.O. Box P M/S 280
Oak Ridge, Tenn 37830

Introduction:

Once upon a time in the beginning when TSC wrote FLEX they figured that four drives would be plenty and so they designed FLEX "from the ground up to handle only four drives." This put a severe restriction on those of us who want to support five inch, eight inch, Winchester, and virtual disk drives all at once. This article presents one way around that problem which requires no modification to FLEX at all. In addition the complete source for a virtual disk driver and formatter is included to illustrate the use of these routines. These are written in 6809 and could be adaptable to 6800 code with some effort.

Background:

When I first decided to attempt the additions to my system I DISASSEMBLED FLEX, expanded the drive tables in the FMS, and modified the initialization for this table. It all seemed like it should work except that it didn't. After many frustrating hours on this I finally decided there must be a better way. After all I didn't want to repeat this process for all six versions of FLEX for which I was responsible. In addition almost every command supplied checked to see that the requested drive was less than three and exited if not. That would mean that every command would have to be DISASSEMBLED and modified. There had to be a better way.

Then one night the light shone upon me and I was illumined as to the better way! I didn't really need more than four drives at any one time. All I needed was a drive splitter and mapper to make the assignments. This would allow me to map the various drives into the FLEX drives. Just one problem, however: the drivers got the drive assignments from FCB 3,X which would be the FLEX assignments and not my assignments. This means that the drive splitter would have to update the drive in 3,X before the call to the driver and restore it upon return. This first approach did not work on my GIMIX FLEX because GIMIX drivers call all their drive routines with a JSR call to the same location which then uses the return address to locate the proper routine after an initial setup. In addition, they call other drive routines through the FLEX JMP table from within some of their drive routines. This caused a jumbled mess of the proper disk assignment.

Eventually I settled upon a system which I believe will solve all of these problems and allow any number of drivers with any number of drives in each to be accessed in FLEX. The only restriction is that a total of four drives may be accessed at any one time. This is accomplished by equating the four FLEX drives with any of the attached drives with the SET command.

The virtual disk drivers were written to take advantage of the extra memory that my OS9 level II uses. It can be mapped to drive 0 so that disk searches begin here and any programs frequently accessed will be loaded without any seek delays in the hardware. You must have the extended addressing enabled on the processor board and all memory boards in the system. This may mean modifying any old memory boards that do not support the extended addressing. You may need only one additional address line decoded for this purpose. Virtual disks make excellent EXTENSIONS to memory in virtual arrays without the time CONSUMING delays of conventional drives.

The drive splitter:

The drive splitter as I call it is primarily based on three routines -- DRIVE, DISKIO, & DRVRTN. Of the nine (ten listed but FLEX does not use the tenth which is seek track) routines in the disk drivers, four directly reference the drive number in FCB 3,X. These are the restore head, select drive, check drive ready, and quick check drive ready. The routine DRIVE handles these cases. The FLEX drive number is pulled from th FCB and stored in the variable GDRIVE for global drive and a nest counter (NESTCT) is incremented to keep track of nested calls so the FLEX drive can be restored only at the proper time in DRVRTN. In addition the stack is set up to return the driver call to the routine DRVRTN (drive return) which checks NESTCT and restores the FCB drive number only on exit to FLEX. The routine DRIVE falls through to the routine DISKIO.

DISKIO is the primary splitter for the drivers. It handles the first three disk routines (read sector, write sector, and verify sector) directly as well as completing the DRIVE routine as discussed in the last paragraph. This routine is responsible for searching the drive equate table DIOVEC for the current drive assignments and then selecting the proper drive routine required.
This it does in part by using a JSR in the FLEX vector table instead of a JMP. This JSR puts the "return" address on the stack as a marker of the entry point which is later deleted in DISKIO after it is used.

The table DIOVEC contains two ENTRIES for each of the FLEX four drives. The first byte is a number which CORRESPONDS to the entry order in the table TYPTBL. For the program listed, type zero CORRESPONDS to the the GIMIX drive table which uses the drivers already present in FLEX through an intermediate JSR table. Type one as assigned at startup as FLEX #3, corresponds to the virtual drivers. Type two if defined would occupy the next definition block in TYPTBL.

The two remaining routines (initialize and warm start) should be handled separately in that these routines should call all the drivers initialize routines and warm start routines respectively. NOTE: The FLEX entries in the vector table at $DE00 should NOT be JSR's but JMP's for these two vectors since they are treated separately and should do a normal return to FLEX.

The table TYPTBL contains all of the definitions for the various drivers present in the system. Each definition consists of four bytes. The first two bytes are a pointer to the respective jump vector teble. The next byte tells the system the largest local drive number allowed. This local drive number is the number associated with the disk driver itself. That is each set of disk drivers starts counting at zero and continues up to its highest drive. The last byte is an upper case ASCII letter which identifies this particular driver. In the example D refers to the current FLEX drivers and V refers to the virtual driver. When expanding this table one only needs to enter as many entries as he desires and the program should handle the number of entries automatically.

The vector jump table and disk driver routines for each driver should adhere to the same requirements that FLEX describes in it's Adaptation Guide. It may be more convenient, however, to get the local drive number from LDRIVE instead of FCB 3,X if you are writing your own drivers. One word may be said here for using the drivers already present in your version of FLEX. The disk vector table at $DE00 must be overwritten to point to the drive splitter routines. Therefore all that is NECESSARY is to recreate this original vector table somewhere else in memory and point to it in the TYPTBL table.

One word of caution here! If you are using TSC or SWTPC FLEX, then this table simply needs to be copied. If you are using GIMIX FLEX then this becomes a bit more complicated. They use a JSR instead of a JMP in their table and they vector all entries to a common routine which does some initialization before continuing to the proper routine. The return address on the stack put there from the JSR is used to select the proper routine. In this case a different "JSR" table is used, which is listed in the source. This table jumps to a different routine that patches the the return address on the stack to make it look like it came from the FLEX table at $DE00 and then jumps to the routine that was called for in the original table. This address is listed for both the G-28 5" double density FLEX and the GIMIX OMA FLEX. If yours is another version of GIMIX FLEX, then check for the proper address.

Driver assignments:

The driver equates are done with the SET command. A word of caution here -- this SET command should be memory resident since it might be possible to map out all of the disks which had this command on them and one would not be able to recover easily from this situation. This also eliminates a need to copy this command on most every media. This SET command also assigns the system and work drives. If the letter S or W precedes the disk assignments then the system and/or work drives will be assigned to all. If the letter S or W follows a drive equate then only that drive will be assigned. This replaces the FLEX ASN utility. The drive assignments have a number of formats that can be used since the NEXT routine which gets the next character in the command line filters all undefined characters below a '1. This means that the command SET S 0=D3,3=D1W,2=V0 which is equivalent to SET S0D33D1W2V0 will assign SYSTEM to all, FLEX drive #0 to D3 (drivers with ID "D" local drive #3), FLEX drive #3 to D1 and make this the work drive, and FLEX drive #2 to the virtual driver, while leaving FLEX drive #1 alone. The routine CHECK will look for duplicates which could screw up the way FLEX handles the SIRs, and flags the duplicates by setting the high order bit in the DIOVEC table local drive number. This will allow the current assignment to print but will not allow access to these drives until they are remedied by a new set command. This assignment error will show up every time SET prints the results. The suppress print option will be overridden at the first encounter of an error and processing of the command line will cease.

FLEX maintains an internal drive data table which consists of four ENTRIES (one for each drive) of six bytes each (two for track and sector of first free sector, two for last free sector, and two for the number of free sectors) for a total of 24 bytes. This table is located at $D41D in 6809 FLEX ($7815 in MiniFLEX, & $7815 in 6800 FLEX). When the first of these bytes is zero FLEX will assume that drive data is missing, and will read the SIR for that drive. Therefore, if you are going to reassign drives within a running program which is WRITING to any of the disks, you will have to zero out this first byte in this table to force a read of the SIR for that disk. This zeroing should only be done on a disk which has all files closed before reassigning the drive. If, on the other hand, you only assign these drives from the FLEX prompt, then there is nothing to worry about since the warm start of FLEX automatically closes all open files.

THE VIRTUAL DISK DRIVER

Anyone who has extended memory addressing enabled on his/her system and supports more than the 56K can use the virtual disk driver and assign all of the memory beyond that which FLEX uses to the virtual disk. The primary advantage here is speed. If your programs use any virtual arrays or you are constantly calling in a few programs or you just have one disk then this virtual disk may be just the cat's meow for you.

This driver turned out to be extremely simple to write once the operation of the DAT was understood. One simply maps in the sector of interest to page zero and copies the sector to the FCB. If it turns out that the FCB might interfere with this memory assignment then page two is used instead. The second page is used instead of page one because the FCB could conceivably lie on the boundary of page zero and one.

In addition the only driver routines of any significance are the read and write routines. All others are meaningless since there is only one drive and its always ready. There is one minor point to mention here, and that is that the virtual warm start routine patches a minor flaw in TSC's FLEX by setting the direct page to zero. The stylograph editor that I use changes the DP and TSC's PASCAL uses the DP but does not initialize it. This patch will prevent the system from bombing in these cases. All of you who use GIMIX FLEX can ignore this point since they have already done the same thing in their drivers.

VIRTUAL DISK DRIVE FORMATTER

The virtual disk, as in all disks, needs to be formatted before use in order for FLEX to handle it properly. If you have a battery backup memory, then besides the initial time, this only needs to be done when it becomes too fragmented. If you do not have

a backup memory then the formatting will have to be done every time you turn the power back on. The formatter for this driver is supplied for this purpose. Both the formatter and the drivers assume that your memory is all good and does not check for bad blocks and does not verify writes.

The variable MAXTRK in FORMATV should match the variable of the same name in the drivers. The table VDAT in the drivers should map the available memory on your system. The example assumes two 32K pages on one and two. Any arrangement is possible and the size is handled automatically by the assembler. The only requirement is that this table match your usable memory. (Obviously, the top 8k on any page is unusable on the SWTPC and GIMIX systems since this is reserved for IO.)

Where do you put it?

In my SWTPC systems this whole package, along with my video drivers, fits very nicely in a 2K RAM on the CPU card at $F000. This leaves all my user memory below $C000 available. On my Gimix systems, the drivers and drive splitters fit very nicely in the AUTOBOOT Prom and everything else fits into into the remaining space on the 1K CPU RAM at $E450-$E6FF. If all else fails, you can always move MEMEND down and stick it there.

Where do I sign?

You can get a copy from me at the above address by sending me a Flex formatted disk (5" or 8") with a postage-paid return mailer. I will pay return postage if you send some software (in source form with some instructions) that you think may be of use to me on the same disk.

```
### FLEX DISK DRIVE EXPANDER ####
## SWTPC DAT VERSION FOR VIRTUAL DISK DRIVE #####
# DISKIO SOURCE
# SELECT PROPER DRIVERS FROM REQUESTED FLEX DRIVE NUMBER

##############################################################
#   Written by Matt Scudiere
#   Union Carbide Corp. / Nuclear Division
#   P.O. Box P  M/S 280
#   Oak Ridge, Tenn 37830
#   (615) 576-7684 9-5 EDST
#
#  For free copy on disk send FLEX formatted
#   disk ( 5" or 8") to above with postage
#   on return mailer.  (Includes source for
#   Priam SMART interface disk drivers 10Mb
#   & 64Mb partitioned drives.)
#  I would appreciate any software that you
#  might deem appropriate in trade.
##############################################################

F000  MEMORY  SET   $F000     SWTPC BOARDS MAY ADD A 2K RAM HERE
                  * GIMIX CPU BOARDS CAN USE $E450-$E6FF FREE RAM
                  * FOR MOST OF THIS CODE IF DESIRED.

DE00  DRVTBL  EQU   $DE00

                  * WORK VARIABLES FOR DRIVES
F000                ORG   MEMORY

F000 F222         VTABLE  FDB   VECVIR    VECTOR TABLE 4 REFERENCE
F002 F24C         VWRITE  FDB   WRITV     WRITE ROUTINE VECTOR 4 FORMATER
F004 02           NDRVRS  FCB   DRVTOT    # OF DRIVERS LISTED IN TYPTBL
F005 00           GDRIVE  FCB   0         GLOBAL (SYS DEFINED) DRV
F006 00           LDRIVE  FCB   0         LOCAL DRV OF RESPEC DSK DRVR
F007 FF           RESTCT  FCB   -1        ENTRY FLAG 4 DRIVE % DISKIO
F008 00           DRIVE1  FCB   0         TEMP WORK SPACE

F009              DIOVEC
                  * INITIAL ASSIGNMENTS
                  * DRIVE TYPE, LOCAL DRV # (REG 4 NON-EXISTANT)
F009 00 00              FCB   0,0       00 GLOBAL DRIVE
F00B 00 01              FCB   0,1       01 "      "
F00D 00 02              FCB   0,2       02 "      "
F00F 01 00              FCB   1,0       03 "      "

                  **** ALL CODE FROM THIS POINT ON IS ROMABLE *****

                  * NOTE: ID IN TYPTBL MUST NOT = 'S' OR 'M'
F011              TYPTBL
                  * DRIVE TYPE 00
F011 F285               FDB   VECGIM    VECTOR FOR GIMIX
F013 03                 FCB   3         HIGHEST LOCAL DRV # ALLOWED
F014 44                 FCC   'D'       ID (MUST BE UPPER CASE)
                  * DRIVE TYPE 01
F015 F222               FDB   VECVIR    VECTO TABLE 4 VIRTUAL DRIVES
F017 00                 FCB   0         ONLY ONE DRIVE
F018 56                 FCC   'V'       ID IS "V" FOR VIRT DISK
                  * INSERT ANY OTHER DRIVE DESCRIPTORS HERE
0002              DRVTOT  EQU   (*-TYPTBL)/4

CC12              USRCMD  EQU   $CC12

F019              CMDTBL
F019 53 45 54 00        FCC   'SET',0
F01D F020               FDB   SET
F01F 00                 FCB   0

                  * "SET" COMMAND FOR RESETTING DRIVE TYPE AND LOCAL DRIVE
                  *     NUMBER FOR EACH OF THE 4 FLEX DRIVES
                  *
                  * SET 0=V0 1=D0 2=01 3=02 ?
                  * NOTE ="S, COMMAS, & SPACES ARE OPTIONAL AND IGNORED
                  * WILL EQUATE FLEX DRIVE 0 TO VIRTUAL DRIVE 0
                  *                         1 TO REGULAR DISK DRIVE 0
                  *                         3 TO REGULAR DISK DRIVE 1
                  *                       & 4 TO REGULAR DISK DRIVE 2
                  * ? - WILL CAUSE PRINT OF CURRENT STATUS (DEFAULT)
                  *     ( USE WHEN PRECEDING ; IN STARTUP FILE IF PRINT DESIRED)
                  * / - WILL SUPPRESS PRINT OF CURRENT STATUS
                  * ; - WILL SUPPRESS PRINT IFF NO ? PRESENT
                  * NOTE - SCAN STOPS AT FIRST ENCOUNTER OF ?,/,;,(CR)
                  *
                  * "S" AND "M" WILL ASSIGN SYSTEM AND WORK DRIVES
                  *
                  * EG.
                  *  IF S APPEARS BE4 DISK ASSIGNMENT WILL SET SYS TO ALL
                  *  IF S APPEARS AFTER DISK ASSGN'MT WILL SET SYS TO THAT DR
                  *  SAME FOR M - WORK ASSIGNMENT
                  *
                  * +++SET S 0=V0
                  * WILL EQUATE THE SYSTEM DRIVE TO "ALL" AND
                  *      EQUATE FLEX DRIVE 0 TO VIRTUAL DRIVE 0 AND PRINT
                  *
                  * +++SET 0V0S 3=M /
                  * WILL EQUATE THE VIRTUAL DRIVE TO FLEX DR 00 AND
                  *      EQUATE THE WORK DRIVE TO PREVIOUSLY DEFINED
                  *       FLEX 03 WITH NO PRINT
                  *
                  * +++SET
                  * WILL RESET THE PROCESSOR DP REGISTER TO 0 AND PRINT
                  *

CD03              WARMS   SET   $CD03
CD4F              CKSUM   SET   $CD4F
CD00              INCH2   SET   $CD00
CD13              SWITCH2 SET   $CD13
CD27              XXTCHR  SET   $CD27
CD18              PUTCHR  SET   $CD18
CD1E              PSTRNG  SET   $CD1E
CD24              PCRLF   SET   $CD24
CC02              TTYEQL  SET   $CC02
CC0B              SYSDRV  SET   $CC0B
CC0C              WRKDRV  SET   $CC0C
CC11              LSTERM  SET   $CC11
CC14              LOFFMT  SET   $CC14

                  * SET COMMAND
F020              SET
F020 4F                 CLRA
F021 1F  8B             TFR   A,DP      RESET DP
F023 B6  CC11           LDA   LSTERM
F026 81  CC02           CMPA  TTYEQL
F029 1027 00A7          LBEQ  FLEX
F02D              SET0
F02D 8D  XX             BSR   NEXT      SET NEXT CHAR
F02F 81  53             CMPA  #'S       SET SYSTEM DRIVE TO ALL?
F031 26  07             BNE   SET1
```

```
F033 B6 FF        LDA   #$FF
F035 B7 CC#B      STA   SYSDRV
F038 2# F3        BRA   SET#
F03A          SE11
F03A 81 57        CMPA  #'W       SET WORK DRIVE TO ALL?
F03C 26 #7        BNE   SET2
F03E B6 FF        LDA   #$FF
F040 B7 CC#C      STA   WKDRV
F043 2# E8        BRA   SET#
F045          SET2
F045 8# 30        SUBA  #'#       CONVERT TO BINARY
F047 81 #3        CMPA  #3        HIGHEST ALLOWED
F049 1#22 #08E    LBHI  SETERR    OOPS TOO BIG
F04D B7 F#0#      STA   DRIVET    SAVE IN TEMP
F050 B9 59        BSR   NEXT      GET NEXT CHAR
F052 81 53        CMPA  #'S       SYSTEM DRIVE FLAG
F054 27 36        BEQ   STSYS     YUP
F056 81 57        CMPA  #'W       WORK DRV FLAG
F058 27 3E        BEQ   STWK      YUP
F05A 10BE F#14    LDY   #TYPTBL+3 PT TO ALLOWED VALUES
F05E 5F           CLRB            INIT COUNT#
F05F          SI1
F05F A1 A4        CMPA  ,Y        CK NEXT ENTRY
F061 27 #A        BEQ   ST7       YUP MATCH
F063 31 24        LEAY  4,Y       4 ITEM / ENTRY
F065 5C           INCB            NO - SET CNT 4 NXT TRY
F066 F1 F#04      CMPB  #DRVRS    CK 4 END OF TBL
F069 25 F4        BLO   SI1       NOPE
F06B 2# 6E        BRA   SETERR    NOT FOUND
F06D          ST7
F06D B8 3C        BSR   NEXT
F06F B# 30        SUBA  #'#       CONVERT TO BINARY
F071 A1 3F        CMPA  -1,Y      TOO BIG ?
F073 22 66        BHI   SETERR    YUP
F075 34 #2        PSHS  A         SAVE LOCAL DRIVE #
F077 B6 F#0#      LDA   DRIVET    GET GLOBAL DRIVE
F07A 48           ASLA            X2 PER ENTRY
F07B 8E F#09      LDX   #DIOVEC   TABLE OF DRIVER DEFN'S
F07E 30 86        LEAX  A,X       GET ABSOL ADDR
F080 35 #2        PULS  A         GET DRIVER # LOCAL DR #
F082 1E 89        EXG   A,B
F084 ED 84        STD   ,X        UPDATE TABLE

F086 8D 23        BSR   NEXT      ASSIGN TO SYSTEM DRIVE?
F088 81 53        CMPA  #'S
F08A 26 #8        BNE   ST#
F08C          STSYS
F08C B6 F#0#      LDA   DRIVET
F08F B7 CC#9      STA   SYSDRV
F092 2# #C        BRA   SET
F094          ST#
F094 81 57        CMPA  #'W       ASSIGN TO WORK DRIVE?
F096 26 #B        BNE   ST9
F098          STWK
F098 B6 F#0#      LDA   DRIVET
F09B B7 CC#C      STA   WKDRV
F09E 2# #8        BRA   SET       CHECK FOR NEXT ENTRY
F0A# 
F0A# BE CC14      LDX   LBFPNT    BACK UP LINE BUFFER POINTER
F0A3 30 1F        LEAX  -1,X
F0A5 BF CC14      STX   LBFPNT
F0A8 16 FF75      LBRA  SET       CHECK FOR NEXT ENTRY
                  * GET NEXT CHAR ON CMD LINE & FILTER
F0AB          NEXT
F0AB 8D CD27      JSR   NXTCHR    NEXT CHAR IN BUFF
F0AE 81 #D        CMPA  #$D       <CR>?
F0B# 27 7B        BEQ   PRINTS    YUP DONE
F0B2 81 CC#2      CMPA  ?IVEC#    BREAK IN LINE?
F0B5 27 1B        BEQ   JMPWRM    YUP
F0B7 81 2F        CMPA  #'/       SUPPRESS PRINT
F0B9 27 17        BEQ   JMPWRM
F0BB 81 3F        CMPA  #'?       FLAG SUMMARY PRINT?
F0BD 27 63        BEQ   PRTNTS    YUP
F0BF 81 3#        CMPA  #'#       SKIP SP,COMMA,ETC
F0C1 2D E8        BLT   NEXT      YUP - SKIP
F0C3 81 3D        CMPA  #'=       FILTER '=' TOO
F0C5 27 E4        BEQ   NEXT      IF '='
F0C7 81 61        CMPA  #'a       lower case?
F0C9 25 #6        BLO   NXTR
F0CB 81 7A        CMPA  #'z
F0CD 22 #2        BHI   NXTR
F0CF 8# 2#        SUBA  #'a-'A    MAKE UPPER
F0D1          NXTR
F0D1 39           RTS             ANYTHING ELSE GETS RETURNED

                  * THIS LINE IS NOT REALLY NEEDED SINCE FLEX RESETS STACK
F0D2          JMPWRM
F0D2 32 62        LEAS  2,S       CLEAN UP RTS ON STAK
F0D4          FLEX
F0D4 8D #D        BSR   CHECK     ANY DUPLICATION ERRORS?
F0D6 26 4C        BNE   PRINTL    YES:
F0D8 7E CD03      JMP   WARMS     THAT'S ALL FOLKS

F0DB          SETERR
F0DB 8E F195      LDX   #MSG      ERROR MESSAGE
F0DE BD CD1E      JSR   PSTRNG    PRINT IT
F0E1 2# 41        BRA   PRINTL

                  * THIS ROUTINE CHECKS 4 DUPLICATES
                  CHECK
F0E3
F0E3 7F F#0#      CLR   DRIVET    NOW ERROR FLAG
F0E6 8E F#0A      LDX   #DIOVEC+1 SET TO CLR ALL ERR FLGS
F0E9 C6 #4        LDB   #4        LOOP COUNT
F0EB          CK1
F0EB A6 84        LDA   ,X
F0ED 84 7F        ANDA  #$7F
F0EF A7 84        STA   ,X
F0F1 3# #2        LEAX  2,X
F0F3 5A           DECB
F0F4 26 F5        BNE   CK1
F0F6 8E F#09      LDX   #DIOVEC   START AT BEGIN OF TBL
F0F9          CK2
F0F9 EC #1        LDD   ,X++
F0FB 1F 12        TFR   X,Y
F0FD 8C F#11      CMPX  #DIOVEC+#
F100 27 1C        BEQ   CK#
F102          CK4
F102 10A3 A4      CMPD  ,Y
F105 26 #D        BNE   CK6
F107 A6 21        LDA   1,Y
F109 8A #8        ORA   #$#8      SET BOTH HI BITS
F10B A7 21        STA   1,Y
F10D CA #8        ORB   #$#8
F10F E7 1F        STB   -1,X
F111 7C F#0#      INC   DRIVET    ERROR CNT
F114
F114 31 22        LEAY  2,Y
F116 10BC F#11    CMPY  #DIOVEC+#
F11A 26 E6        BNE   CK4
F11C 2# #8        BRA   CK2
F11E          CK#
F11E 7D F#0#      TST   DRIVET    ANY ERRORS?
F121 39           RTS

                  * STATUS PRINT OUT OF CURRENT DISK EQUATES
F122              PRINTS
                  * THIS LINE IS NOT REALLY NEEDED SINCE FLEX RESETS STACK
F122 32 62        LEAS  2,S       CLEAN UP RTS ON STAK
F124              PRINTL
F124 8D #D        BSR   CHECK     VERIFY NO DUPLICATES
F126 8E F#09      LDX   #DIOVEC   PT TO DEFN'S
F129 BD CD24      JSR   PCRLF     NEW LINE
F12C 5F           CLRB            CLEAR DRIVE CTR
F12D          PLOOP
F12D 1F #9        TFR   B,A       COPY TO A 4 PRINT
F12F 8B 3#        ADDA  #'#       CONVERT TO ASCII
F131 8D CD1#      JSR   PUTCHR    PRINT IT
F134 86 3D        LDA   #'=       SET = TO PRT
F136 8D CD1#      JSR   PUTCHR    DO IT
F139 A6 B9        LDA   ,X+       GET DRIVER TYPE
F13B 10BE F#14    LDY   #TYPTBL+3 PT TO CHAR TYPE TBL
F13F 48           ASLA
F140 48           ASLA            #4 ITEMS / ENTRY
F141 A6 A6        LDA   A,Y       GET CHAR TO PRT
F143 8D CD1#      JSR   PUTCHR    PRINT IT
F146 A6 84        LDA   ,X        GET LOCAL DRV DEFN
F148 84 7F        ANDA  #$7F      STRIP ERROR BIT
F14A 8B 3#        ADDA  #'#       CONVERT TO ASCII
F14C 8D CD1#      JSR   PUTCHR    PRINT IT
F14F 86 2#        LDA   #'        INSERT SP
F151 8D CD1#      JSR   PUTCHR
F154 7D CC#B      TST   SYSDRV    SEE IF ALL
F157 2B #9        BMI   WRITS     YUP
F159 F1 CC#9      CMPB  SYSDRV    IS THIS THE SYS
F15C 27 #4        BEQ   WRITS
F15E 86 2#        LDA   #'        NO PRT SP
F160 2# #2        BRA   PLI       DO IT
F162          WRITS
F162 86 53        LDA   #'S       PRT S 4 SYS DR
F164          PLI
F164 8D CD16      JSR   PUTCHR    OUTPUT
F167 7D CC#C      TST   WKDRV     SEE IF ALL
F16A 2B #9        BMI   WRITN     YUP
F16C F1 CC#C      CMPB  WKDRV     IS THIS THE WK DRV
F16F 27 #4        BEQ   WRITN     YUP
```

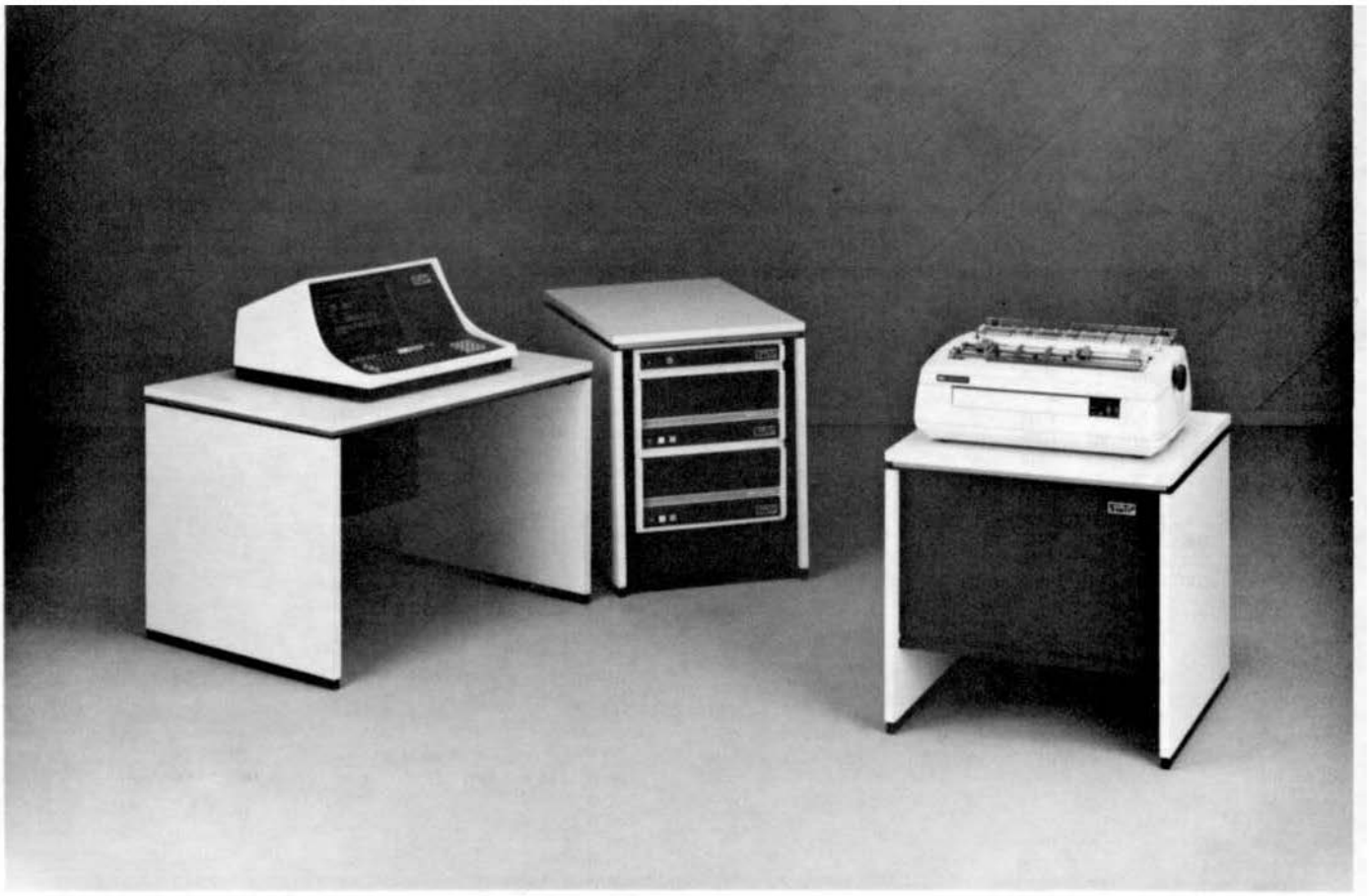
```
F171 86    20          LDA    #'          NO PRT SP
F173 20    02          BRA    PL2         DO IT
F175              WRITW
F175 86    57          LDA    #'W         PRT W 4 WR DRV
F177              PL2
F177 BD    C018        JSR    PUTCHR      DO IT
F17A A6    80          LDA    ,X+         RECOVER & DUP CK
F17C 85    80          BITA   #$80        SET IF ERROR
F17E 27    0A          BEQ    PL3         BRA IF NO ERR
F180 34    16          PSHS   D,X         SAVE
F182 8E    F195        LDX    #MSG
F185 BD    C01E        JSR    PSTRNG      FLAG DUPLICATIONS
F188 35    16          PULS   D,X         RECOVER
F18A              PL3
F18A BD    C024        JSR    PCRLF       NEW LINE
F18D 5C                INCB               GO TO NEXT GLOBAL DRIVE
F18E C1    04          CMPB   #4          END OF LIST?
F190 25    90          BLO    PLOOP       NO - LOOP
F192 7E    C003        JMP    WARMS       YES  DONE

                  * ERROR MESSAGE
F195              MSG
F195 20 2A 2A 20       FCC    ' ** ERR **',D,4
F199 45 52 52 20
F19D 2A 2A 0D 04

                  * DISKIO SOURCE
                  * SELECT PROPER DRIVERS FROM REQUESTED DRIVE NUMBER

                  * ENTRY 4 DRIVE REFERENCE DRIVE ROUTINES
F1A1              DRIVE
F1A1 34    C6          PSHS   D,U,PC      DUMMY U & PC PLACE HOLDERS
F1A3 7C    F007        INC    NESTCT      KEEP TRK OF ENTRIES & NESTED CALLS
F1A6 26    05          BNE    DI01
F1A8 A6    03          LDA    3,X         GET GLOBAL DRIVE #
F1AA B7    F005        STA    GDRIVE      SAVE IT
F1AD              DI01
F1AD EC    66          LDD    6,S         MOVE DUMMY RTN ADDR
F1AF ED    62          STD    2,S         PAST RTN ADDR
F1B1 AF    66          STX    6,S         TEMP STORAGE 4 X
F1B3 CC    F1F9        LDD    #DRVRTN     ADJUST RTN FROM DRIVERS
F1B6 ED    64          STD    4,S         PUT IN STACK
F1B8 35    06          PULS   D           RESTORE & FALL THROUGH

                  * ENTRY FOR NON-DRIVE REFERENCE DRIVE ROUTINES
F1BA              DISKIO
F1BA 34    26          PSHS   D,Y         SAVE
F1BC B6    F005        LDA    GDRIVE      GET GLOBAL DRIVE
F1BF 81    03          CMPA   #3          TOP LIMIT FLEX DEFN
F1C1 22    46          BHI    ERROR
F1C3 48                ASLA               2 BYTES /ENTRY
F1C4 108E  F009        LDY    #DIOVEC     DRIVE TABLE
F1C8 EC    A6          LDD    A,Y         DRV TYPE & LOGICAL DRIVE
F1CA B1    F004        CMPA   NODVRS      # OF DRIVE TYPES
F1CD 22    3A          BHI    ERROR       EXCEEDED TABLE
F1CF 5D                TSTB               CK 4 NON EXISTANT DRV
F1D0 2B    37          BMI    ERROR
F1D2 F7    F006        STB    LDRIVE      UPDATE LOCAL DRV
F1D5 7D    F007        TST    NESTCT      DRIVE 4 ACCESS?
F1D8 26    02          BNE    DI02        NO - SKIP
F1DA E7    03          STB    3,X         SET LOCAL DR # IN FCB 4 DRIVERS
F1DC              DI02
F1DC 48                ASLA               X2
F1DD 48                ASLA               X4 ITEMS / ENTRY
F1DE 108E  F011        LDY    #TYPTBL     PT TO VECTOR TABLE
F1E2 31    A6          LEAY   A,Y         PT Y TO DRIVE VECTOR
F1E4 B6    F006        LDA    LDRIVE      RECOVER 4 CK
F1E7 A1    22          CMPA   2,Y         CK 4 LDRIVE TOO LARGE
F1E9 22    1E          BHI    ERROR
F1EB 10AE  A4          LDY    ,Y          GET ACTUAL ADDR OF VECTOR
F1EE E6    65          LDB    5,S         RTN ADDR
F1F0 C0    03          SUBB   #3          CORRECT FOR OFFSET
F1F2 31    A5          LEAY   B,Y         ADD TO Y FOR JUMP ADDR
F1F4 10AF  64          STY    4,S         JUMP VECTOR ON STACK
F1F7 35    A6          PULS   D,Y,PC      RESTORE & JUMP

                  * RETURN FROM DRIVE REF DISK ROUTINES
                  * ( NON-DRIVE REFERENCE ROUTINES RETURN DIRECTLY TO CALLER)
F1F9              DRVRTN
F1F9 35    10          PULS   X           RESTORE X
F1FB 34    01          PSHS   CC          SAVE ERROR FLAG
F1FD 7A    F007        DEC    NESTCT      DEC NEXT COUNT
F1FF 24    05          BCC    DI04        BRA IF NOT RETURNING TO FLEX
F202 B6    F005        LDA    GDRIVE      RESTORE GLOBAL
F205 A7    03          STA    3,X         DRIVE IN FCB
F207              DI04
F207 35    81          PULS   CC,PC       BACK TO FLEX
```

```
                  * BAD CALL EXIT
F209              ERROR
F209 35    26          PULS   D,Y         RESTORE
F20B 32    62          LEAS   2,S         CLEAN UP RTS FROM ENTRY
F20D C6    0F          LDB    #$0F        NO SUCH DRIVE
F20F 1A    01          SEC                FLAG ERROR
F211 39                RTS                DONE

                  ********************************
                  * LIB VIRTDISK.SRC



                  **************************************************
                  * VIRTUAL DISK DRIVERS
                  * SET DAT ADDR 4 AVAIL MEM (NOTE INVERTED 2ND NIBBLE)
                  * ANY # OF SECTIONS ALLOWED - PROGRAM WILL HANDLE OK
                  * LOW 32 K ON PAGE 1 & 2 DEFINITION FOR VDAT
F212              VDAT
F212 1F 1E 1D 1C       FCB    $1F,$1E,$1D,$1C,$1B,$1A,$19,$18
F216 1B 1A 19 18
F21A 2F 2E 2D 2C       FCB    $2F,$2E,$2D,$2C,$2B,$2A,$29,$28
F21E 2B 2A 29 28
           000F   MAXTRK  EQU  *-VDAT-1
           DFD0   LRARAM  EQU  $DFD0
           FFF0   DATRAM  EQU  $FFF0

                  * VIRTUAL DISK VECTOR TABLE
F222              VECVIR
F222 16    001D        LBRA   READV
F225 16    0024        LBRA   WRITV
F228 16    007A        LBRA   VERIFV
F22B 16    0077        LBRA   RESTOV
F22E 16    0074        LBRA   DRIVEV
F231 16    0071        LBRA   CHKRDV
F234 16    006E        LBRA   QUICKV
F237 16    0068        LBRA   DINITV
F23A 16    0065        LBRA   DWARMV
F23D 16    0065        LBRA   DSEEKV

                  * FORMAT OF MEM IS 16 "SECTORS"/"TRACK"
                  * HI NIBBLE IS "TRACK" (EXTENDED ADDR: A16-A20)
                  * LO NIBBLE IS "SECTOR" (A12-A15)
F240              READV
F240 34    01          PSHS   CC
F242 1A    10          SEI
F244 8D    2F          BSR    SETDAT
F246 1E    12          EXG    X,Y
F248 8D    1D          BSR    COPYXY
F24A 20    08          BRA    RESDAT


F24C              WRITV
F24C 34    01          PSHS   CC
F24E 1A    10          SEI
F250 8D    23          BSR    SETDAT
F252 8D    13          BSR    COPYXY

                  * RESTORE DAT
F254              RESDAT
F254 B6    DFD0        LDA    LRARAM      RESTORE SECTION 0
F257 B7    FFF0        STA    DATRAM
F25A B6    DFD2        LDA    LRARAM+2    RESTORE SECTION 2
F25D B7    FFF2        STA    DATRAM+2
F260 35    01          PULS   CC
F262 C6    00          LDB    #0
F264 1C    FE          CLC
F266 39                RTS
                  * COPY BLOCK FROM X-PTR TO Y-PTR
                  * DESTROYS D,X,Y
F267              COPYXY
F267 C6    80          LDB    #128
F269 34    04          PSHS   B           COUNTER ON STACK
F26B              LOOP
F26B EC    81          LDD    ,X++
F26D ED    A1          STD    ,Y++
F26F 6A    E4          DEC    ,S
F271 26    F8          BNE    LOOP
F273 35    84          PULS   B,PC

                  * SET THE DAT TO MAP REQUIRED SECTOR INTO PAGE 0 IF
                  * THIS DOES NOT INTERFERE WITH FCB ELSE MAP TO
                  * PAGE 2 AND POINT Y TO THE MAPPED SECTOR.
F275              SETDAT
F275 5A                DECB               SECTOR NOW 0-15 (FROM 1-16)
F276 C1    0F          CMPB   #15         SEC/TRK
F278 22    1F          BHI    SEEKER
```

```
F27A 81    0F              CMPA   #MAXTRK
F27C 22    1B              BHI    SEEKER
F27E 108E F212             LDY    #VDAT      ADD EXT
F282 A6    A6              LDA    A,Y
F284 8C    1000            CMPX   #$1000     SEE IF INTERFERE W DATA MEM
F287 25    05              BLO    SD1
                   * USE PG 0 SINCE FCB DATA IS NOT IN THIS AREA
F289 B7    FFF0            STA    DATRAM     SET DAT PG 0
F28C 20    05              BRA    SD2
                   * USE PG 2 SINCE FCB DATA IS IN PG 0
F28E               SD1
F28E B7    FFF2            STA    DATRAM+2
F291 C8    20              ADDB   #$20       SET 4 2ND SECTION USED
F293               SD2
F293 1F    98              TFR    B,A        COPY SECTOR (0-TS) TO A
F295 5F                    CLRB
F296 1F    02              TFR    D,Y        SET Y TO MEM ADD OF SECTOR
F298 39                    RTS

                   * ERROR IN SEEK (INVALID TRK OR SECTOR)
F299               SEEKER
F299 32    62              LEAS   2,S        CLEAN UP RTS
F29B 35    01              PULS   CC         RESTORE IRQ
F29D C6    10              LDB    #$10       SEEK ERRO
F29F 1A    01              SEC               FLAG ERROR
F2A1 39                    RTS

F2A2               DWARMV
                   * THIS CODE IS UNNECCESSARY ON GIMIX FLEX SYSTEMS
F2A2 5F                    CLRB
F2A3 1F    9B              TFR    B,DP       CLEAR DP TO PATCH FLEX
F2A5               VERIFV
F2A5               RESTOV
F2A5               DRIVEV
F2A5               [illegible]
F2A5               QUICKV
F2A5               DINITV
F2A5               DSEEKV
F2A5 5F                    CLRB              $ CARRY 4 NO ERRORS
F2A6 39                    RTS

                   * END VIRT DISK DRIVER
                   **********************


                   * SELECT ONE
             DE59  GIMDSK  SET    $DE59      GIMIX DMA FLEX
             DE56  GIMDSK  SET    $DE56      GIMIX G-28 FLEX

F2A7               GVECTR
F2A7 34    06              PSHS   D          SAVE
F2A9 EC    62              LDD    2,S        GET RTN ADDR
F2AB C3    EB40            ADDD   #$DE00-VECGIN CONVERT TO $DE00 ORG
F2AE ED    62              STD    2,S        PUT BACK ON STACK
F2B0 35    06              PULS   D          RESTORE
F2B2 7E    DE56            JMP    GIMDSK


                   * VECTOR TABLE FOR GIMIX (SEE TEXT)
F2B5               VECGIN
F2B5 BD    F2A7            JSR    GVECTR
F2B8 BD    F2A7            JSR    GVECTR
F2BB BD    F2A7            JSR    GVECTR
F2BE BD    F2A7            JSR    GVECTR
F2C1 BD    F2A7            JSR    GVECTR
F2C4 BD    F2A7            JSR    GVECTR
F2C7 BD    F2A7            JSR    GVECTR
F2CA               GCOLD
F2CA BD    F2A7            JSR    GVECTR
F2CD               GWARM
F2CD BD    F2A7            JSR    GVECTR
F2D0 BD    F2A7            JSR    GVECTR


F2D3 8D    F5      COLD    BSR    GCOLD
F2D5 BD    F237            JSR    VECVIR+$15 INCLUDE NEW DRIVE(S)
                   * ADD ADDITIONAL DRIVE REFERENCES HERE
F2D8 39                    RTS

F2D9 8D    F2      WARMD   BSR    GWARM
F2DB BD    F23A            JSR    VECVIR+$18 INCLUDE NEW DRIVE(S)
                   * ADD ADDITIONAL DRIVE REFERENCES HERE
F2DE 39                    RTS


         F2DF  ZZZZZZ  EQU    *          FLAG LAST ASEMBLED ADDRESS
```

```
CC12                       ORG    USRCMD
CC12 F019                  FDB    CMDTBL     ADD COMMAND TABLE ENTRY

                   *************************************
                   * LAST TO LOAD IS NEW VECTOR TABLE

C100                       ORG    $C100      FLEX .CMD AREA

                   * INSTALL DISK DRIVER VECTOR TABLE
C100               START
                   * CK FOR JMP INSTRUCTION IN DISK DRIVE TBL
                   *  IF NOT JMP THEN MUST BE GIMIX JSR CALL
                   * WHICH SHOULD BE CORRECTED IN THIS SOURCE
C100 86    7E              LDA    #$7E       'JMP'
C102 C6    09              LDB    #9         # ENTRIES IN TBLE
C104 8E    DE00            LDX    #DRVTBL
C107 A1    84      SU2     CMPA   ,X
C109 26    15              BNE    CPYNEW     SKIP IF JSR
C10B 30    03              LEAX   3,X
C10D 5A                    DECB
C10E 26    F7              BNE    SU2
                   * COPY OLD TSC/SWTPC TABLE TO VECGIN
C110 8E    DE00            LDX    #DRVTBL
C113 108E F2B5             LDY    #VECGIN
C117 C6    1B              LDB    #(ENDTBL-JMPTBL)
C119 A6    80      SU4     LDA    ,X+
C11B A7    A0              STA    ,Y+
C11D 5A                    DECB
C11E 26    F9              BNE    SU4
                   * COPY NEW TABLE TO DE00
C120               CPYNEW
C120 8E    C133            LDX    #JMPTBL
C123 108E DE00             LDY    #DRVTBL
C127 C6    1B              LDB    #(ENDTBL-JMPTBL)
C129 A6    80      SU6     LDA    ,X+
C12B A7    A0              STA    ,Y+
C12D 5A                    DECB
C12E 26    F9              BNE    SU6
C130 7E    F020            JMP    SET


C133               JMPTBL
C133 BD    F1BA            JSR    DISKIO     READ SECTOR
C136 BD    F1BA            JSR    DISKIO     WRITE SECTOR
C139 BD    F1BA            JSR    DISKIO     VERIFY SECTOR
C13C BD    F1A1            JSR    DRIVE      RESTORE HEAD
C13F BD    F1A1            JSR    DRIVE      SELECT DRIVE
C142 BD    F1A1            JSR    DRIVE      CHECK DRIVE READY
C145 BD    F1A1            JSR    DRIVE      QUICK CK DRIVE READY
                   * NOTE: NEXT TWO ARE JMP INSTEAD OF JSR
C148 7E    F2D3            JMP    COLD       COLD START
C14B 7E    F2D9            JMP    WARMD      WARM START
                   * THIS LAST ENTRY IS NOT USED BY FLEX AND MAY
                   * NOT BE PRESENT IN YOUR VERSION OF FLEX
                   *  INCLUDE ONLY IF REFFERENCED BY ADDED
                   *  DRIVER(S) TO THE FLEX TABLE AT $DE18

                   * JSR DISKIO SEEK TRACK
         C14E  ENDTBL  EQU    *


                           END    START
0 ERROR(S) DETECTED

SYMBOL TABLE:

CHECK  F0E3   CK1    F0EB   CK2    F0F9   CK4    F102   CK6    F114
CK8    F11E   CKRDYV F2A5   CKSTAT C04F   CMDTBL F019   COLD   F2D3
COPYXX F267   CPYNEW C120   DATRAM FFF0   DINITV F2A5   DIO1   F1A0
DIO2   F10C   DIO4   F207   DIOVEC F009   DISKIO F1BA   DRIVE  F1A1
DRIVET F000   DRIVEV F2A5   DRVRTN F1F9   DRVTBL DE00   DRVTDT 0002
DSEEKV F2A5   DWARMV F2A2   ENDTBL C14E   ERROR  F209   FLEX   F004
GCOLD  F2CA   GDRIVE F005   GIMDSK DE56   GVECTR F2A7   GWARM  F2CD
INCH2  C00D   JMPTBL C133   JMPWRM F002   LOFFNT CC14   LDRIVE F006
LOOP   F260   [illegible]
NS6    F195   NDRVES F004   NESTCT F007   NEXT   FAAD   NXTCHR CD27
NXTR   F0E1   OUTCH2 C013   PCRLF  C024   PL1    F164   PL2    F177
PL3    F18A   PLOOP  F120   PRINTL F124   PRINTS F122   PSTRNG C01E
PUTCHR C018   QUICKV F2A5   READV  F240   RESDAT F254   RESTOV F2A5
SD1    F28E   SD2    F293   SEEKER F299   SET    F020   SET0   F020
SET1   F03A   SET2   F045   SETDAT F275   SETERR F00B   ST1    F05F
ST7    F06D   ST8    F094   ST9    F0A0   START  C100   STSYS  F08C
STWK   F090   SU2    C107   SU4    C119   SU6    C129   SYSDRV CC0B
TTYEOL CC02   TYPTBL F011   USRCMD CC12   VDAT   F212   VECGIN F2B5
VECVIR F222   VERIFV F2A5   VTABLE F000   VWRITE F002   WARMD  F2D9
WARMS  C003   WKDRV  CC0C   WRITS  F162   WRITV  F24C   WRITW  F175
ZZZZZZ F2DF
```

```
                 * NOTE START TRK & SEC MUST BE >= 00,05
           0000  STRTRK  EQU   0
           0006  STRTSC  EQU   6          * 2 BLOCKS 4 DIR (20 FILES)

C100                     ORG   $C100      .CMD
                 * VIRTUAL DISK FORMATTER


           000F  MAXTRK  SET   $F         *** MUST MATCH SAME IN VIRT DRIVERS ****


           F002  VWRITE  SET   $F002      INDIRECT ADDR FOR VIRTUAL WRITE

           CC0E  DATE    SET   $CC0E      FLEX DATE
           CD1E  PSTRNG  SET   $CD1E
           CD18  PUTCHR  SET   $CD18
           CD39  OUTDEC  SET   $CD39
           CD42  GETHEX  SET   $CD42
           CD15  GETCHR  SET   $CD15
           CD24  PCRLF   SET   $CD24
           CD1B  INBUF   SET   $CD1B
           CD2D  GETFIL  SET   $CD2D
           CD48  INDEC   SET   $CD48
           D406  FMS     SET   $D406
           D403  FMSCLS  SET   $D403
           CD3C  OUT2HS  SET   $CD3C
           CD03  WARMS   SET   $CD03


C100 20    18    FMTVIR  BRA   FMT1
C102 01                  FCB   1          VERSION 0
C103             PROMPT
C103 BD    CD1E          JSR   PSTRNG     PRINT STRING
C106 BD    CD15          JSR   GETCHR     GET ANSWER
C109 84    5F            ANDA  #$5F       MAKE UPPER CASE
C10B 81    59            CMPA  #'Y        SEE IF YES
C10D 39                  RTS

C10E             EXIT
C10E 8E    C1CE          LDX   #ABORT
C111 BD    CD1E          JSR   PSTRNG
C114 BD    D403          JSR   FMSCLS
C117 7E    CD03          JMP   WARMS

C11A             FMT1
C11A 8E    C19A          LDX   #SURES
C11D 8D    E4            BSR   PROMPT
C11F 26    ED            BNE   EXIT


                 * CLEAR WK SPACE & INIT FORWARD LINKS
C121 8E    C225          LDX   #WORK
C124 5F                  CLRB             COUNTER =256
C125             FMT2
C125 6F    80            CLR   ,X+
C127 5A                  DECB
C128 26    FB            BNE   FMT2

C12A CC    0002          LDD   #2         1ST FWD LINK
C12D 8E    C225          LDX   #WORK      RESET
C130 ED    84            STD   0,X
C132 CC    0001          LDD   #1         RESET TO START

C135             FMTGO
C135 AD    9F F002       JSR   [VWRITE]   WRITE IT OUT
C139 25    5A            BCS   WOUPS      WRITE ERROR
C13B 8E    C225          LDX   #WORK      RESET
C13E EC    84            LDD   ,X         TK & SEC FORWARD LINK
C140 34    06            PSHS  D          PUT ON STACK 4 NXT WT
                 * COMPUTE NEXT SECTOR
C142 C1    10            CMPB  #MAXSEC    LAST SEC ON TRK
C144 25    02            BLO   NXTS1      NO INC SEC
C146 4C                  INCA             YES INC TRK
C147 5F                  CLRB             RESET SEC
C148             NXTS1

C148 5C                  INCB             BUMP
C149 ED    84            STD   ,X         SET FWD LINK
C14B 35    06            PULS  D          RECOVER CUR TRK & SEC
C14D 1083  C21C          CMPD  LSTSEC     AT END OF DISK YET
C151 26    E2            BNE   FMTGO      NOT YET CONTINUE
                 * CLR LAST SEC ON DISK FWD LNK
C153 6F    84            CLR   ,X
C155 6F    01            CLR   1,X
C157 FC    C21C          LDD   LSTSEC     LAST SEC ON "DISK"
C15A AD    9F F002       JSR   [VWRITE]
C15E 25    35            BCS   WOUPS

                 * INIT DIRECTORY (0,5 TO 0,FSTSEC-1)
                 *  SET LAST SEC IN DIR FWD LINK = 0
C160 FC    C21A          LDD   FSTSEC     LAST SEC ON TRK 0
C163 C0    02            SUBB  #2
C165 2A    05            BPL   DA1
C167 C6    0F            LDB   #MAXSEC-1
C169 4A                  DECA
C16A             DA1
C16A 5C                  INCB
C16B 8E    C225          LDX   #WORK
C16E AD    9F F002       JSR   [VWRITE]   CLEAR FWD LINK
C172 25    21            BCS   WOUPS

                 * SET SYS INFO RECORD
C174 8E    C1FD          LDX   #SIRBLK
C177 FC    CC0E          LDD   DATE
C17A FD    C220          STD   SDATE
C17D B6    CC10          LDA   DATE+2
C180 B7    C222          STA   SDATE+2
C183 CC    0003          LDD   #3         TRK 0, SEC 3
C186 AD    9F F002       JSR   [VWRITE]
C18A 25    09            BCS   WOUPS

C18C 8E    C1E9          LDX   #COMPLT
C18F             DONE
C18F BD    CD1E          JSR   PSTRNG

C192 7E    CD03          JMP   WARMS

C195             WOUPS
C195 8E    C1CE          LDX   #ABORT
C198 20    F5            BRA   DONE



                 * PRINT STRINGS
C19A 41 72 65 20 SURES   FCC   'Are you sure you want to format'
C19E 79 6F 75 20
C1A2 73 75 72 65
C1A6 20 79 6F 75
C1AA 20 77 61 6E
C1AE 74 20 74 6F
C1B2 20 66 6F 72
C1B6 6D 61 74
C1B9 20 74 68 65         FCC   ' the virtual disk ? ',4
C1BD 20 76 69 72
C1C1 74 75 61 6C
C1C5 20 64 69 73
C1C9 6B 20 3F 20
C1CD 04
C1CE 2A 2A 2A 20 ABORT   FCC   '*** Formatting aborted ***',4
C1D2 46 6F 72 6D
C1D6 61 74 74 69
C1DA 6E 67 20 61
C1DE 62 6F 72 74
C1E2 65 64 20 2A
C1E6 2A 2A 04
C1E9 46 6F 72 6D COMPLT  FCC   'Formatting complete',4
C1ED 61 74 74 69
C1F1 6E 67 20 63
C1F5 6F 6D 70 6C
C1F9 65 74 65 04
                 * WORK SPACE

           0010  MAXSEC  EQU   16         MAX # OF "SEC/TRK"

                 * SYSTEM INFO BLOCK
           C1FD  SIRBLK  EQU   *
C1FD 00 00       FWDLNK  FCB   0,0
C1FF 0000 0000           FDB   0,0,0,0,0,0,0
C203 0000 0000
C207 0000 0000
C20B 0000
C20D 56 69 72 74 VOLNAM  FCC   'Virtual-Mem'
C211 75 61 6C 2D
C215 4D 65 6D
C218 0000        VOLNUM  FDB   0
C21A 00 06       FSTSEC  FCB   STRTRK,STRTSC
C21C 0F 10       LSTSEC  FCB   MAXTRK,MAXSEC
C21E 00FD        TOTSEC  FDB   (MAXTRK-STRTRK+1)*(MAXSEC-STRTSC+1
C220 00 00 00    SDATE   FCB   0,0,0
C223 0F          NXTRK   FCB   MAXTRK
C224 10          NXSEC   FCB   MAXSEC
                 * SIR OVERFLOWS INTO WORK BLOCK

                 * WORK BLOCK USED FOR INITIALIZATION
C225             WORK    RMB   256

           C325  ZZZFMT  EQU   *
```



THE COMPLETE BUSINESS SYSTEM

+Multiuser+Highly Expandable+Cost Effective

S+ THE CONCEPT

The S+ system is a modular computer system in which all portions of the hardware and software are designed to work together in the most efficient way possible. An S+ single user system with floppy disk storage is a competitive and cost effective entry level system. Unlike most other small computers being sold as "personal", or "small business" machines, the S+ system may be expanded to maximum capabilities using this same hardware and software. You cannot end up with a DEAD END system that cannot be expanded and whose software is not compatible with larger machines. A basic S+ system may be expanded to thirty-two users, a megabyte of main memory and hundreds of megabytes of hard disk storage by simply plugging in, or connecting the desired upgrade equipment.

TOTAL DESIGN—Hardware and Software

The S+ system is an integrated hardware and software design. The two complement and enhance each other in this system. The UniFLEX® operating system used in the S+ systems is patterned after the Bell Laboratories UNIX® operating system, one of the most admired and widely used operating systems in the world. Instead of being an afterthought, the software is part of the design of the S+ system. You can be sure that with this approach that all parts of the computer operate with maximum efficiency and cost effectiveness.

THE CENTRAL PROCESSOR

The basic S+ system is configured with 256K bytes of memory and can be expanded to more than 1 million bytes. An efficient and fast hardware memory management system is used to allocate the available memory among the users on a dynamic basis. As little as 8K bytes, or the entire memory—if needed—can be used by any individual user. This makes it possible to run very large programs on the system, but it also uses no more memory than necessary for a particular job. The increase in cost effectiveness of this system over crude and outdated bank switching arrangements is dramatic.

The central processor runs in both user and supervisor states. It can detect and reject a defective user program. It is impossible for a user program to go bad and stop the entire system, as can happen quite easily in less sophisticated systems.

Task switching is accomplished by use of a multiple map RAM memory, with sixty-four individual task maps. Each task can access from 4 to 64 K-bytes of memory. Multiple tasks may be used in programs that require more than 64K bytes of memory for execution. When a task is completed the memory is automatically released for other use.

SOFTWARE

The S+ operating system, UniFLEX® is a multiuser, multitasking operating system based on the UNIX® operating system that has been used for many years on Digital Equipment Corp. PDP-11 series minicomputers. It is considered one of the most sophisticated and "user friendly" operating systems available. Variations of UNIX® are rapidly becoming standard on mini and larger microcomputers.

A large variety of languages are available for use with the system. These include FORTRAN, COBOL, BASIC, and Pascal. Word processing packages are also available to give you full text processing capability on the system.

Applications programs are available in large quantities in many fields. This includes general business, medical, dental, veterinary, library and real estate management; plus others. Since the system is multiuser it can also be connected to cash registers to produce a point-ofsale terminal system combined with the computer. The possibilities for application of this system are endless.

THE I/O SYSTEM

The S+ system is totally interrupt driven. All terminal and printer I/O devices connect to an I/O bus separate from the main bus. Up to thirty-two separate devices may be connected to the I/O bus at any one time. If I/O activity is great enough to cause an unacceptable slowdown in system operation, a separate I/O processor can be installed in the system. This plug-in option removes all I/O handling overhead from the main processor and allows operation of up to thirty-two external devices at 9,600 baud. Without an integrated total design, as in the S+ system, it would become impractical to use a UNIX®type operating system in a situation with heavy terminal I/O activity.

DISK STORAGE

A wide range of disk storage capacity is available for the S+ system, from 2.5 M-byte floppy disks to an 80 M-byte Winchester and many sizes between. All disk controllers use direct memory access (DMA) type operations to maximize data transfer and to minimize overhead on the main processor. The Winchester disks also use intelligent controllers along with DMA transfers to preserve the performance that these type devices are capable of giving. Without this distributed intelligence the system performance would be greatly degraded. The UniFLEX®operating system is designed to work at maximum efficiency with this type disk system. The data transfer rates achieved by this combination rival those of large minicomputers.

COMMUNICATIONS

A high speed local network communications system is available to interconnect S+ systems. The VIABUS® network will allow communication between systems at data rates of over 400K baud. Such a system makes it possible to share data between local systems in an efficient and low-cost manner.

AVAILABLE SOON

Tape backup—20M-Byte in less than 15 minutes on a standard ¼ inch cartridge.

Mini-Wini—5 and 10 M-Byte Winchesters—5¼ inch package. Winchester performance, for smaller systems in a small package. UniFLEX® compatible design.

Large Capacity—190 and 340 M-Byte Winchesters, plus SMD cartridge drives.

UniFLEX is a registered trademark of Technical Systems Consultants, Inc.

UNIX is a registered trademark of Bell Labs.

VIABUS is a registered trademark of Southwest Technical Products Corporation.

```
0 ERROR(S) DETECTED

SYMBOL TABLE:

ABORT  C1CE   COMPLT C1E9   DATE   CC0E   DONE   C19F   ERT    C16A
EXIT   C10E   FMS    D406   FMSCLS D403   FMT1   C11A   FMT2   C125
FMTGO  C135   FMTVIR C100   FSTSEC C21A   FWDLNK C1F0   GETCHR CD15
GETFIL CD2D   GETNEX CD42   HISEC  C224   HITRK  C223   INBUF  C01B
INDEC  CD48   LSTSEC C21C   MAXSEC 0010   MAXTRK 000F   NXTS1  C148
OUT2HS CD3C   OUTDEC CD39   PCRLF  CD24   PROMPT C103   PSTRNG CD1E
PUTCHR CD18   SDATE  C220   SINBLK C1FD   STDTRK 0000   STDTSC 0006
SURES  C19A   TOTSEC C21E   VOLNAM C200   VOLNUM C21B   WRITE  F002
WARMS  CD03   WORK   C225   WOUPS  C195   ZZZFMT C325
```

```
***************************************************
* VIRTUAL DISK DRIVERS
* SET DAT ADDR 4 AVAIL MEM (NOTE INVERTED 2ND NIBBLE)
* ANY # OF SECTIONS ALLOWED - PROGRAM WILL HANDLE OK
* LOW 32 K ON PAGE 1 & 2 DEFINITION FOR VDAT
VDAT
 FCB $1F,$1E,$1D,$1C,$1B,$1A,$19,$18
 FCB $2F,$2E,$2D,$2C,$2B,$2A,$29,$28
MAXTRK EQU *-VDAT-1
LRARAM EQU $DFD0
DATRAM EQU $FFF0

* VIRTUAL DISK VECTOR TABLE
VECVIR
 LBRA READV
 LBRA WRITV
 LBRA VERIFV
 LBRA RESTOV
 LBRA DRIVEV
 LBRA CKRDYV
 LBRA QUICKV
 LBRA DINITV
 LBRA DWARMV
 LBRA DSEEKV

* FORMAT OF MEM IS 16 "SECTORS"/"TRACK"
* HI NIBBLE IS "TRACK" (EXTENDED ADDR: A16-A20)
* LO NIBBLE IS "SECTOR" (A12-A15)
READV
 PSHS CC
 SEI
 BSR SETDAT
 EXG X,Y
 BSR COPYXY
 BRA RESDAT

WRITV
 PSHS CC
 SEI
 BSR SETDAT
 BSR COPYXY

* RESTORE DAT
RESDAT
 LDA LRARAM RESTORE SECTION 0
 STA DATRAM
 LDA LRARAM+2 RESTORE SECTION 2
 STA DATRAM+2
 PULS CC
 LDB #0
 CLC
 RTS

* COPY BLOCK FROM X-PTR TO Y-PTR
* DESTROYS D,X,Y
COPYXY
 LDB #128
 PSHS B COUNTER ON STACK
LOOP
 LDD ,X++
 STD ,Y++
 DEC ,S
 BNE LOOP
 PULS B,PC

* SET THE DAT TO MAP REQUIRED SECTOR INTO PAGE 0 IF
* THIS DOES NOT INTERFERE WITH FCB ELSE MAP TO
* PAGE 2 AND POINT Y TO THE MAPPED SECTOR.
SETDAT
 DECB SECTOR NOW 0-15 (FROM 1-16)
 CMPB #15 SEC/TRK
 BHI SEEKER
 CMPA #MAXTRK
 BHI SEEKER
 LDX #VDAT ADD EXT
 LDA A,X
 CMPX #$1000 SEE IF INTERFERE W DATA MEM
 BLO SD1
* USE PG 0 SINCE FCB DATA IS NOT IN THIS AREA
 STA DATRAM SET DAT PG 0
 BRA SD2
* USE PG 2 SINCE FCB DATA IS IN PG 0
SD1
 STA DATRAM+2
 ADDB #$20 SET 4 2ND SECTION USED
SD2
 TFR B,A COPY SECTOR (0-15) TO A
 CLRB
 TFR D,Y SET Y TO MEM ADD OF SECTOR
 RTS

* ERROR IN SEEK (INVALID TRK OR SECTOR)
SEEKER
 LEAS 2,S CLEAN UP RTS
 PULS CC RESTORE IRQ
 LDB #$10 SEEK ERRO
 SEC FLAG ERROR
 RTS

DWARMV
* THIS CODE IS UNNECCESSARY ON GIMIX FLEX SYSTEMS
 CLRB
 TFR B,DP CLEAR DP TO PATCH FLEX
VERIFV
RESTOV
DRIVEV
CKRDYV
QUICKV
DINITV
DSEEKV
 CLRB & CARRY 4 NO ERRORS
 RTS

* END VIRT DISK DRIVER
**********************
```

Basic To Pascal

BASIC TO PASCAL
By: Ronald W. Anderson

A BOOK REVIEW

"If you understand the fundamentals of BASIC language programming...If you're looking for a way to get faster results and more versatility in your programming practice, then it's time you discovered Pascal!"

Sound familiar? It should, the style at least, it is from the backcover of a new TAB book, titled **From BASIC to Pascal**, by 'our' Ron Anderson who is well known for his **FLEX User Notes**, a 68 Micro Journal regular.

In the very early days of 6800 kit builders, way back in the mid 70's, when there were no 'Apples', 'Ataries', 'TRS 80s', 'IBM Personal Computers', 'etc.' there was Standard S50 Bus computers, mostly kits, and a new and powerful disk operating system 'FLEX™'. FLEX was not the first 68XX disk system, Motorola had one, but much like its' 'blind' editor and assembler it left much to be desired.

Then with the first disk system for the Standard S50 Bus from SWTPc came DOS by Bob Ulterwyk. Fast as 'greased lightning but (we later learned) employed little error checking. Funny thing, we never seemed to notice. DOS, in its' day, like Bobs' BASIC was truly 'state of the art'. But as we and 6800 systems matured the little things, before ignored, began to show their need. Then came FLEX, actually miniFLEX and a fellow by the name of **Ron Anderson**.

Ron had been around during the earlier times, but with the advent of FLEX, and the way now clear to really do 'computer things' like (and in some respects better) than the 'biggies', Rons' talents appeared in one of, if not the first 6800 publication, **FLEX User Notes**, a small but very useful and popular newsletter, which he founded and published.

When 68 Micro Journal was founded the FLEX **User Notes** was the **only** source of accurate and current 68XX information going. By this time Ron was 'up to his ears' in making a living in the explosive microcomputer marketplace. Soon after 68 Micro Journal began publishing, an agreement was made between Ron and myself to absorb Rons' effort into 68 Micro Journal. Over the years literally thousands of 68XX users have told me that without Rons way of making 'hard things simple' they would never have been able to learn to program and operate their Standard S50 Bus computer. For that we all owe Ron a well deserved **THANKS - RON!**

As you can well imagine then, a book on programming by 'Ron Anderson', especially for users of languages that run on 'our' computers, is a **MUST** for the Standard S50 Bus computer user.

BASIC to Pascal is written in typical Anderson fashion. Simple to understand, yet **complete** for any programmers needs, beginner or pro. Over 300 pages of material that makes the transition from BASIC to Pascal a snap. An easy way to gain programming speed, memory space, and versatility.

Ron has a unique way of explaining things, his effort in this book reflects the Anderson touch. The beginner is led 'by the hand' from simple but useful BASIC and Pascal functions and program examples to larger and more useful functions and programming examples. I use the word 'useful' here in a very positive way. In many books of this type one is either 'heaped upon' with a barrage of useless and trivial material or 'burdened to death' with complexity to no end. Some appear to have been written as ego trips for the author or another way **of selling pulp paper. Well, NOT here!!!**

Ron Anderson has managed to compress into these 300 or so pages enough information to allow anyone the least bit acquainted with BASIC (especially the TSC Extended BASIC) to progress to the point that he or she can soon be writing and using Pascal programs. All without the usually painful process of 'relearning'. This is a book of transition as well as teaching. It makes the task seem more like play. It can be no better put than to say, "It is typical Ron Anderson".

Most examples, and there are many, are useful functions. No useless exercises in 'flipping bits' to display the authors skills. Written in a building block type methodology that blends learning with doing into pleasant hours of reading and practice.

The subjects are numerous, covering most all areas of programming algorithms for records, strings, pointers and variables. There is also a summary of Pascal syntax, a glossary of terms, comparisons of Pascal compiler code to assembler code, and a look at extensions to several popular Pascal versions available to the Standard S50 Bus user, to mention a few. The reader soon gets the compelling urge to turn on the computer and start doing.

This is one of the better books on programming I have ever read. It is a **MUST** for anyone, Standard S50 Bus or any other, considering Pascal as a programming tool. Even if it had been written by someone I had never heard of before, it would still be the same, one of the best learning tools on my shelf! And if I were selling a version of Pascal this book would be a part of the documentation furnished.

From:

TAB Books Inc
Blue Ridge Summit, PA 17214

DMW - - -

HELP ME !

Need some HELP

Some time back I was approached by another publishing company with a request to consider writing a book on the history of the Standard S50 Bus. Written in a vein of anecdotes of early users and happenings. Matters of little importance to anyone other than those of us who use Standard S50 Bus computers. A combination of tales, long and short of what it was like, way back when.

Well, I have been around for most all of that time and can remember much, but some of you have memories and experiences that would be of interest to all of us, going back to the very first days.

I don't have the time or patience to do a whole book. But I think that most readers would enjoy a column, now and then, recalling some of the happenings we experienced, in those early days. For me to do it right I need some input from you 'oldtimers' on the Standard S50 Bus. I would hope that those who pioneered the early hardware and software days would furnish some insights. Also some of you who also put together some of the early kits could sure help. Together our combined recollections would certainly make interesting reading.

So, if you would please, let me know if you have anything that you believe would make interesting material. I will attempt to combine what I remember with your input and put it to paper, for our readers. All who contribute will be acknowledged.

I will try to start the columns within the next few months, so let me hear soon. If it will take you awhile to get it together, at least let me know if you are going to contribute. This way I can sorta schedule my timing of it.

Any subject matter appreciated, as long as it pertains to the Standard S50 Bus, 68XX or individuals associated with these subjects.

If you are a manufacturer or early day vendor I know that you have a lot of interesting recollections, how about sharing them with all of us. Wouldn't even mind your mentioning the product (free plug), what is important is letting others know about all the fun we had in those days.

Thanks fellows (and gals) hoping we can get something going.

Don - - -

DYNAMITE PLUS

D Y N A M I T E +
(product review)

We recently received two new versions of the original DYNAMITE disassembler. The new versions, called DYNAMITE+ are for the Flex and Uniflex operating systems. Some of the more important new features in DYNAMITE+ are as follows:

1. Improved label file format with easier usage.
2. Expanded boundary declarations.
3. Internal optimizations which reduce disk accesses.
4. A powerful cross reference generator has been supplied.
5. Uniflex utility to convert Flex binary file to Uniflex format.

After disassembling over 50 programs with these new versions, we feel that DYNAMITE+ surpasses many other disassemblers we have used in the past. It's extremely important that a disassembler produce source code that, when reassembled, will precisely match the original binary file. DYNAMITE+ has disassembled programs and utilities ranging from 50 to over 20,000 bytes in length and has always produced correct code.

Newcomers to computing are the first to ask "Why do I need a disassembler?". Well, if you've ever had a program product that didn't do exactly what you wanted or you've wanted to write some assembly language programs/utilities yourself, then you desperately need a disassembler. A disassembler allows you to make minor (or major) changes to ANY program! Perhaps some examples will illustrate just how useful a disassembler can be to your operation.

How about that program that doesn't output correctly for your system? Perhaps you are using a printer at 8 lines per inch, but the program only outputs 58 lines per page. DYNAMITE+ helps fix that in a matter of minutes.

Perhaps you would like to modify some utilities so that printed output to that new typewriter would automatically stop at the end of each page, allow you to insert a new sheet, then await your prompt to continue outputting data. DYNAMITE+ can simplify implementating this feature.

Some characters, notably the $7F, are considered control characters. If a program sends out one of these, our printer considers it a command to delete it's buffer! DYNAMITE+ was used to eliminate this problem.

Want to add support for that new printer, typewriter or plotter to BASIC? Want graphic plot commands for that video display board in BASIC? A disassembler makes these tasks relatively easy and in fact has been used to support IBM selectric typewriters (using correspondence code), Baudot teletypes, hi-speed paper tape readers/punches, video boards, etc. In one particular version of BASIC, a 'morse' command was implemented. It works just like the print command, except that it sends out the data in morse code on a speaker as well as providing a signal for the transmitter's keying relay!

One of the best ways to learn to write in any language is by studying the work of others. Many of you have perused or perhaps studied the Flex and Uniflex programmer's manuals but some of you are still wondering just how to make use of all that information! These are reference manuals, and although they provide excellent examples, sometimes they just aren't enough to get you started. Due to time constraints, we found this to be true with the Uniflex system. Several readings of the complete manual still left us wondering about numerous points. After disassembling over 20 utilities such as 'copy, check, dir, free, mount, list, and path' we now feel very familiar with Uniflex. It's no longer so mysterious or difficult to understand! And having disassembled these utilities has given us a firm foundation in writing Uniflex utilities for ourselves and our customers. We can only say that without a disassembler, we would have procrastinated for many more months before 'getting into' Uniflex. Now we almost feel like seasoned veterans!

Using a disassembler is really quite simple. One merely tells the disassembler to disassemble any given binary file. It can be as simple as that. However, the assembly language code produced by the disassembler, although accurately representing the original binary code, may not be as meaningful as desired. For a simple program patch, it will probably be adequate. But if you intend to make extensive modifications or desire to keep the source text for later reference, then you will want to take additional steps to make the text more meaningful. Let's say you would like to disassemble the 'list' command. First, call up DYNAMITE+ and have it disassemble the utility and produce a listing which shows all ascii characters. This original listing will appear to have many weird commands, label equates, etc. This is because DYNAMITE+ didn't know where any of the ascii strings (error messages, etc) were in the program and disassembled them thinking they were binary instructions. After examining this listing, you will easily be able to identify these ascii strings. In addition, you will usually also notice some areas used for line or page counters, etc. Once again you call up DYNAMITE+. But this time you tell DYNAMITE+ that you want to inform it of these areas you have discovered. After doing same, DYNAMITE+ once again produces a listing. However, this listing is usually quite readable. Now all that remains is to examine this listing and replace some of the labels generated by DYNAMITE+ with labels of your own choosing which you feel are more meaningful. Finally you call up DYNAMITE+ and have it disassemble the file one last time, using the code areas you discovered and using your labels instead of its own. When we are disassembling a program for complete documentation purposes, we usually follow the following course of action with DYNAMITE+.....

1. Produce listing showing all ascii and suppress disk output.
2. Produce listing using ascii/binary boundaries discovered.
3. Produce disk file of text using boundaries and labels discovered.

The above procedure produces a meaningful text file of assembly language source code which can be further enhanced using your favorite editor to add comments, make any necessary changes, etc. Using these techniques we have completely disassembled, commented, and otherwise documented the majority of the Flex system and its utilities and are doing same for Uniflex. Without a disassembler such as DYNAMITE+, this would be an impossible task. With DYNAMITE+ however, it's almost fun to see in how short a time it (and you) can 'blast' apart that utility!

The DYNAMITE+ programs for Flex and Uniflex have so many features that it's hard to adequately cover them in detail. Some of their features are as follows:

1. Disassemble 6800 or 6809 code.
2. Provides standard labels and equates for Flex and Uniflex.
3. Provides standard labels for Mikbug, Swtbug, S-Bug, & Diskbug.
4. Handles system 'sys' calls within Uniflex code.
5. Handles 'text' and 'bss' sections of Uniflex code.
6. Produces a complete cross reference of all labels and/or opcodes.
7. Provides a utility to convert Flex binary files to Uniflex format.

DYNAMITE+ allows you to provide it with data boundaries so that it doesn't try to disassemble ascii strings, constants, variables, and data areas. This may be done interactively at the terminal, or you can build a text file (using your editor) of lines of text which outline to DYNAMITE+ these various boundaries. For example:

A 1000-103F means there are ascii strings in the addresses shown.
S 1050-1073 means these are binary areas in the addresses shown.
L 1100-1107 means there are label FDB's in the addresses shown.
W 1200-1203 means there are word FDB's in the addresses shown.

Another powerful boundary line format specification is.....

<rpc> <type><range> [';'<type><range>...]
where.....
<rpc> is a line repeat count
<type> is the data area type (A/B/C/L/W)
<range> is the range of addresses to be used

The above format appears complex, but in actual use is quite simple. However, we will not discuss it in detail here. suffice to say that is extremely powerful and its greatest use is in specifying command tables, jump tables, and for building DYNAMITE+ command files that are easily modified for subsequent releases of products/utilities which often vary only slightly and only require a minimum of work to once again disassemble.

Since DYNAMITE+ provides standard labels and equates for common monitor and system calls, this means that when a JSR $C003 is encountered in the binary file, DYNAMITE+ will automatically substitute JMP WARMS instead. Therefore, many calls will be converted for you to meaningful labels like INCH, OUTCH, PDATA, GETCHR, PUTCHR, PSTRNG, PCRLF, FMS, and FMSCLS. When processing Uniflex binary files, system calls like 'sys read,bufadr,buflen' or 'sys write,messag,msglen' are also handled with no intervention by you. This saves valuable time and minimizes hours wasted in constantly using the reference manual. These standard labels can also be modified or enhanced by you, and can even be disabled if desired.

One very nice feature of DYNAMITE+ is the ability to specify on the command line any and all options you may wish to invoke. This allowed us to build 'exec' type files containing standard lines of commands which directed DYNAMITE+ to disassemble many programs per run while we attend to other matters. The command line options allowed are.....

1. Print ascii equivalent of code on source line.
2. Interactively prompt user for data area boundaries.
3. Suppress creating a text file on the disk.
4. Expand disassembler output by adding a blank line before
   any line containing a label.
5. generate extra lines of code bytes (prints all object code).
6. Suppress listing.
7. Allow Motorola format for zero offset indexed mode. This option
   recognizes the difference between LDA 0,X and LDA and accounts for a bug in the TSC assembler.
8. Generate line numbers on the lefthand side of the output.
9. Enable pagination. Allows inputting of your title.
10. Disable printing of time and date on header (Uniflex version).
11. Disassemble 6800 code, not 6809 code which is the default.
12. Specify the command file name which holds the data boundaries.
13. Specify the system file name (Uniflex version).
14. Specify the output file name (Uniflex version).
15. Specify the label file for handling system calls, etc.
16. Specify number of lines to be printed per page.

Since a picture is worth a thousand words, what follows is the actual output from a DYNAMITE+ Uniflex disassembly. The only addition to the listing is our comments which appear on lines 17 thru 37.

```
1                       * disassembly by dynamite+ of fj
2
3                               info   Filter to block ^L's
4                               info   (ie, after assembly)
5                               info   from clearing screen.
6                               info   by Scott Schaeferle
7                               info   June 21,1981
8
9                       * system name equates
10
11                0005  term    equ    5
12                000C  read    equ    12
13                000D  write   equ    13
14
15                      * external label equates
16
17                0040  s0040   equ    $0040           i/o buffer
18
19  0000  CC   0000      s0000   ldd    #$0000          select std i/p
20  0003  113F 0C                sys    read,s0040,$0001 read 1 char of i/p
21  0006  1025 0021              lbcs   s002f          if err goto errorx
22  000A  1083 0000              cmpd   #$0000          if not eof
23  000E  26   03                bne    s0013           ,goto ck for ^L
24  0010  113F 05                sys    term            ..else terminate
25  0013  B6   0040      s0013   lda    >s0040          get i/p char
26  0016  81   0C                cmpa   #$0c            if chr not ^L (FF)
27  0018  26   05                bne    s0023           ,go write output
28  001A  86   07                lda    #7              ld substitute char
29  001C  B7   0040              sta    >s0040          store in i/p bufr
30  001F  CC   0001      s0023   ldd    #$0001          select std o/p
31  0022  113F 0D                sys    write,s0040,$0001 write 1 chr of o/p
32  0025  24   D1                bcc    s0000           if no err go read
33  0027  34   06        s002f   pshs   a,b             push error code
34  0029  CC   0001              ldd    #$0001          select err o/p
35  002C  113F 0D                sys    write,s0041,$0006 write Error msg
36  002F  35   06                puls   a,b             pull error code
37  0031  113F 05                sys    term            terminate job
38
39  0034                         org    $0041
40
41  0041  45 72 72 6F  s0041    fcc    'Error!'        Error message
42
43                0000           end    s0000
```

It's easy to imagine having DYNAMITE+ make one final pass on the file using your labels such as readlp equ $0000, chek0c equ $0017, writop equ $0023, errorx equ $002f, and errmsg equ $0041. Try manually substituting these labels into the above listing and see just how convenient a disassembler can be. It can save many man hours which means money in your pocket and satisfied customers! We give the highest rating possible to these two fine products. They performed flawlessly and appear to be 'bug free' in all respects.

The DYNAMITE+ disassemblers are available for the Flex and Uniflex systems. They are sold by Computer Systems Center, 13461 Olive Blvd., Chesterfield, MO. 63017, telephone (314) 576-5020. See their ad in any issue of 68 Micro Journal.

William E. Fisher - - -

BIT BUCKET

PERCOM DROPS OFF STANDARD S50 BUS

I received a call from PERCOM this week informing us that PERCOM has dropped their Standard S50 Bus line of computer equipment.

PERCOM was one of the early 6800 manufacturers, and the late Harold Mauch, PERCOM founder was very active in both software and hardware design.

As there are still users of the PERCOM systems, PERCOM felt that someone might be interested in continuing the PERCOM line. If so interested parties should contact:

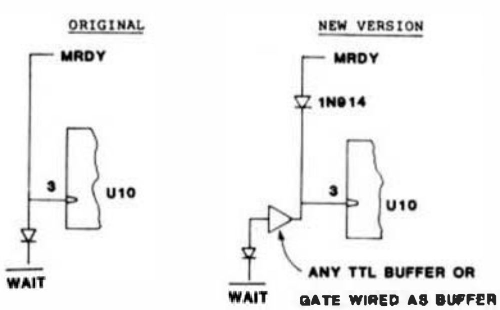
Mr. John Adell Jr.
PERCOM data Company, Inc.
11220 Pagemill Road
Dallas, Texas 75243
A/C (214) 340-7081 ext. 58

Also they have a supply of bare boards for practically all PERCOM products. If you are interested in these, in any quanity, please contact John Adell, at the address above.

DMW - - -

David J. Graves
311A Towne Building/D3
U. of Pa.
Philadelphia, Pa., 19104

There are two improvements which can be made in the MREADY circuit of the Winchester interface I described in the October issue of 68 Micro. One or the other (or both) may be essential in specific systems, and they may improve the reliability of any system. The first is the addition of a buffer to the incoming WAIT signal, and the second is the addition of a diode so that MREADY can be pulled low by devices other than the Winchester interface (a "wired or" configuration). Both improvements have been made in the printed circuit board mentioned in that article in addition to full buffering of all address lines and inclusion of an extended address decoding option for 50C busses. The bare board is available for $40 and an assembled and tested version is $120. For those who want to do it on their own, here are the suggested improvements:



20 October, 1982

Mr. Don Williams Sr.
68 Micro Journal
Hixon, TN, 37343

Dear Don:

First I would like to express my thanks for the numerous help which I have received through your published articles in Micro 68. Two of these, recently published, were invaluable in getting my Stylograph Word Processor going.

For most of the 'Standard Buss' users, there is little or no problem getting commercial software operational. For the rest of us, there are some problems.

This letter deals with some minor modifications required with getting the STYLOGRAPH WORD PROCESSOR working on Creative Micro Systems hardware (Exorciser Buss). Since I had the General Version of of Flex up and running, I assumed that the program would operate as written. When nothing happened, I used the 'GET' command to enter the program into memory. Scanning through the program, I noticed $E004 and that immediately indicated that the program required some address changes for hardware constants. Stylograph makes the task quite easy since the provide the source for the I/O routines. I reassembled the source using the Flex Edit, and Assembler. When I tried the program a second time, I was able to get the menue on the screen, but any subsequent input from the terminal resulted in an incoherent display. This indicated a timing problem.

A word of explanation is required here. CMS has used a divide by 64 instead of 16 so any ACIA Initialize routines must be changed to reflect both the different address ($E3C0) but also the timing. These are:

$E004........................$E3C0
86 03........................86 13
86 11........................86 16

I had taken care of this so that accounted for the manue display but the timing problem remained. Further checking of the printed listing of the assembled I/O source indicated that in this program ACIA interrupts are required and that for my system the line LDA #$91 required a change to LDA #$92. I reassembled and the program worked.

There seemed to be a few quirks with the program however and after attempts, I tried a new approach. Instead of appending the new I/O program to Stylo, I put the original program in memory using the 'GET' command. I had previously located the program on disk with the DISKFIX9 program published in th Aug 82 issue of 68 Micro Journal and now, rather than saving the program after the necessary changes, I corrected the Disk using the 'DISCUSS.CMD' program published in the Feb 82 issue. The changes are as follows:

ADDRESS	CONTENTS	CHANGE
$005A	$E004	$E3C0 • I/O ACIA TO TERMINAL
005C	FFFF	BDAF • HIGH END OF RAM
0060	DFC0	E713 • SYSTEM INTERRUPT
009D	86 91	86 92 • DIVIDE BY 64
00A6	86 13	86 03 • RESET
00AF	86 11	86 16 • DIVIDE BY 64

Having made the changes, the program was called using the STYFIX command; this allows you to modify the program for your terminal and printer. The program now functioned as advertized. It appears that the main body of the program may have been modified and that these mods were not carried to the I/O source program. So far so good, and now for the final test. To do a printout of the text, you need a print program for your printer. The Stylo manual indicates that you can use the FLEX PRINT.SYS routine if it works properly with your printer. The manual suggests that copying this program to the Stylo Disk under the new name STYPRINT.CMD. Doing this took care of activating the printer.

While this letter deals with a specific modification for CREATIVE MICRO SYSTEMS hardware, It should be applicable for any non-standard system using FLEX DOS. Just make the necessary mods at the addresses shown in column I above. For further information concerning the Stylograph Word Processing System I suggest you read the May issue of 68 Micro Journal for the review.

Don, as you can see, the programs published in 68 Micro Journal are not only functional, but tutorial as well.

very truly yours,

Lt/Col Anthony J. Gambarro,
Box 78
E.Sullivan, N.H., 03445

Mr. Don Williams, Sr.
'68 MICRO JOURNAL
5900 Cassandra Smith
P.O. Box 849
Hixon, TN 37343

Don,

I've been a subscriber for about a year now and thought I might finally submit my two cents worth regarding your fine magazine. Let me start by responding to Ron Anderson's "EDITORIAL" in the OCT. issue concerning the "FASTER AND BIGGER IS BETTER" syndrome. I must say that I am in whole hearted agreement. Certainly a need exists for super hi-speed processors but I don't believe it's in the home computer field. I think that most owners using a 1 MHZ 6809 will find that our systems are waiting for us, most of the time. And I'm not sure that the home computer market can justify an 8 or 10 MHZ 68000 at this time. By the way, Ron, if you need a 1 MHZ 6809 CPU card on the SS-50 bus, take a look at Data Systems "68" product line. They haven't forgotten the low budget computerist.

I must say that Bob Nay and Tony Distefano have presented some very useful information to the TRS-80C user. In fact, my only gripe is that the COLOR USER NOTES column has kind of gotten a little heavy with FLEX conversion info. Bob, please don't forget us little guys. As a suggestion, you might try to present some information on routines in the BASIC rom, entry addresses and the like. I sure a lot of this info has become available.

For those who have a need, there is a 6809 one line disassembler, written in position-independent code, presented in the FEB. 1982 issue of BYTE magazine. It's easily adaptable to the TRS-80C if you have an assembler. Don't worry Don, you still have the best 68XX magazine That I've been able to find.

I've taken on a project that I hope you or your readers can be of assistance on. I am attempting to put together an SS-50 bus computer from bare cards. Since I have more time than money, I must do this in a piece by piece process. I have a 6809 CPU card and a backplane (4 SS-50 slots) on the way. I hope to eventually expand it to a system capable of running FLEX. But for the time being, I'll probably use it strictly as a controller to spool my printer and such. The main thing I need is INFORMATION. Things like what monitors are available, what all do I need to be able to install FLEX? How much trouble will I encounter, software wise, if I use a video board and parallel keyboard instead of a terminal. I thought I might be able to use my TRS-80C as a terminal for starters. All of the information that you SS-50 bus guys just take for granted can really be a help to those of us trying to upgrade on a budget. Any and all information would be appreciated.

For those TRS-80C owners who are inclined toward hardware info, I've built an EPROM programmer for mine, that works quite well and can be built quite cheaply. It burns a 2716 in about 110 seconds with a machine language driver and could be easily modified for 2732's and the various 8K Eproms coming on the market. I'll be glad to share this with anyone for an SASE. I expect the MICRO TECHNICAL PRODUCTS 2716 programmer could be easily connected to a wire-wrapped parallel port plugged into the expansion connector. I have also modified some old program paks that I no longer needed to accept the 2K 6116/2016 RAMS. They make good temporary storage and can be write-protected to retain your data if your program crashes. As you may have noticed, some of the earlier RADIO SHACK software for the TRS-80C leaves a lot to be desired. Might as well put those unused program paks to good use. You can also add 8K between $8000 and $9FFF if you don't have Extended Basic installed.

Thanks,
H.E. Villers, Jr.
19240 S.W. 310 Ter.
Homestead, FL 33030

```
0010 REM *** WP.BAS *** REVISED 9/30/82 ***
0011 REM ** RANDY D. KRALLIN **
0012 REM ** 2720 REDWOOD DR **
0013 REM ** APTOS, CA 95003 **
0020 PRINT CHR$(26)
0030 ON ERROR GOTO 900
0040 OPEN "O,PRINT" AS 0
0050 PRINT "* THIS IS A LIMITED WORD PROCESSING PROGRAM.  SIMPLY RESPOND TO"
0060 PRINT "THE PROMPTS FROM THE COMPUTER.  UPPER & LOWER CASE, INDENTATION"
0070 PRINT "AND ALL TYPES OF PUNCTUATION ARE ALLOWABLE.  SELECT ONE OF THE"
0080 PRINT "THREE MAJOR OPERATIONS LISTED BELOW TO BEGIN.  EXIT BY ENTERING"
0090 PRINT "FUNCTION (CONTROL) "C" AT ANY TIME A "?" IS DISPLAYED, EXCEPT'
0100 PRINT 'WHEN IN THE "TYPE" OR "EDIT" MODES.  IN THESE CASES ENTER -"X" AS'
0110 PRINT 'THE FIRST AND ONLY CHARACTER ON A LINE, THEN PRESS "RETURN". *'
0120 PRINT :PRINT:PRINT
0130 INPUT "DO YOU WISH TO TYPE, PRINT OR EDIT IT, P, OR E!",A$
0140 IF A$="T" OR A$="t" GOTO 180
0150 IF A$="P" OR A$="p" GOTO 370
0160 IF A$="E" OR A$="e" GOTO 460
0170 PRINT :PRINT:PRINT "RE-ENTER SELECTION!!!";:GOTO 120
0180 PRINT :PRINT:PRINT:INPUT "ENTER DOCUMENT NAME  (8 LETTERS MAX.)",N$
0190 OPEN NEW N$ AS 1:PRINT:PRINT:PRINT
0200 INPUT "ENTER BEGINNING LINE NUMBER (USE 10)",M
0210 FOR J=M TO M+99
0220 PRINT M;: IF M<10 THEN PRINT " ";
0230 GOSUB 870
0240 INPUT LINE B$:PRINT #1,M;PRINT #1,B$
0250 IF B$="X" THEN CLOSE 0,1:GOTO 330
0260 PRINT #0,B$
0270  M=M+1:IF M<60 GOTO 320
0280 PRINT :PRINT:INPUT "THIS ENDS A NORMAL PAGE.  CONTINUE ON - Y/N",D$
0290 IF D$="Y" OR D$="y" GOTO 320
0300 IF D$="N" OR D$="n" THEN B$="Z":PRINT #1,M;PRINT #1,B$:CLOSE 0,1:GOTO 330
0310 PRINT :PRINT:PRINT "RE-ENTER SELECTION!!!":GOTO 280
0320 NEXT J
0330 PRINT :PRINT:PRINT: INPUT "DO YOU WISH TO RETAIN THIS DOCUMENT - Y/N",Z$
0340 IF Z$="Y" OR Z$="y" GOTO 20
0350 IF Z$="N" OR Z$="n" GOTO 990
0360 PRINT :PRINT:PRINT "RE-ENTER SELECTION!!!":GOTO 330
0370 PRINT :PRINT:PRINT:INPUT 'IS PRINTER READY?  TYPE "Y" FOR YES',C$
0380 IF C$<>"Y" AND C$<>"y" GOTO 370
0390 PRINT :PRINT:PRINT: INPUT "ENTER DOCUMENT NAME",N$:OPEN OLD N$ AS 1
0400 PRINT :PRINT:PRINT
0410 INPUT #1,M: INPUT LINE #1,B$
0420 PRINT M;: IF M<10 THEN PRINT " ";
0430 PRINT B$:IF B$="X" THEN CLOSE 0,1 ELSE GOTO 450
0440 GOTO 330
0450 PRINT #0,B$:GOTO 410
0460 PRINT :PRINT:PRINT
0470 INPUT "ENTER NAME OF DOCUMENT TO BE EDITED",N$:OPEN OLD N$ AS 1
0480 INPUT #1,M: INPUT LINE #1,B$
0490 PRINT M;: IF M<10 THEN PRINT " ";
0500 PRINT B$: IF B$="X" THEN CLOSE 1:GOTO 520
0510 GOTO 480
0520 PRINT :PRINT:PRINT:OPEN OLD N$ AS 1:OPEN NEW "TEMP" AS 2
0530 INPUT "ENTER NUMBER OF LINE TO BE EDITED",L
0540 INPUT #1,M: INPUT LINE #1,B$: IF M<L GOTO 570
0550 PRINT #2,M;:PRINT #2,B$:PRINT M;: IF M<10 THEN PRINT " ";
0560 PRINT B$:GOTO 540
0570 PRINT M;: IF M<10 THEN PRINT " ";
0580 PRINT B$:PRINT:PRINT
0590 PRINT "IS THIS LINE TO BE CHANGED, OR IS TEXT TO BE INSERTED"
0600 INPUT "ENTER THIS LINE?  C = CHANGE    I = INSERTION",X$
0610 IF X$="C" OR X$="c" GOTO 640
0620 IF X$="I" OR X$="i" GOTO 780
0630 PRINT :PRINT:PRINT "RE-ENTER SELECTION!!!",:GOTO 590
0640 PRINT M;: IF M<10 THEN PRINT " ";
0650 GOSUB 870
0660 INPUT LINE B$:PRINT #2,M:PRINT #2,B$:GOSUB 710
0670 IF B$="X" GOTO 20
0680 INPUT #1,M: INPUT LINE #1,B$:PRINT #2,M:PRINT #2,B$:GOSUB 710
0690 IF B$="X" GOTO 740
0700 GOTO 680
0710 IF B$="X" THEN CLOSE 1,2
0720 IF B$="X" THEN KILL N$+".DAT":RENAME "TEMP.DAT",N$+".DAT"
0730 RETURN
0740 PRINT :PRINT:PRINT:INPUT "ANYMORE ON THIS DOCUMENT - Y/N",Y$
0750 IF Y$="Y" OR Y$="y" THEN OPEN OLD N$ AS 1:GOTO 680
0760 IF Y$="N" OR Y$="n" GOTO 20
0770 PRINT :PRINT:PRINT "RE-ENTER SELECTION!!!":GOTO 740
0780 PRINT #2,M;PRINT #2,B$
0790 PRINT :PRINT:PRINT:INPUT "ENTER LINE NUMBER AS A DECIMAL",M
0800 PRINT M;:GOSUB 870
0810 INPUT LINE B$:PRINT #2,M:PRINT #2,B$:PRINT:PRINT:PRINT
0820 INPUT "ANOTHER INSERTION IN SEQUENCE - Y/N",Y$
0830 IF Y$="Y" OR Y$="y" GOTO 790
0840 INPUT #1,M: INPUT LINE #1,B$:PRINT #2,M:PRINT #2,B$
0850 GOSUB 710
0860 IF B$="X" THEN GOTO 740 ELSE GOTO 640
0870 FOR J=1 TO 69:PRINT "-";
0880 NEXT J
0890 PRINT "-":PRINT "  ";:RETURN
0900 IF ERR/2>30 AND ERR/2>8 AND ERR/2>4 GOTO 970
0910 IF ERR/2>30 GOTO 930
0920 PRINT :PRINT:PRINT "PLEASE RE-ENTER YOUR SELECTION!";:RESUME
0930 IF ERR/2>8 GOTO 950
0940  M=M+1:B$="X":PRINT #2,M:PRINT #2,B$:RESUME
0950 PRINT :PRINT:PRINT "COULDN'T FIND DOCUMENT, TRY AGAIN!"
0960 IF A$="P" OR A$="p" THEN RESUME 390 ELSE RESUME 460
0970 PRINT :PRINT:PRINT "YOU HAVE CAUSED AN ERROR HALTING PROGRAM"!
0980 PRINT " EXECUTION!  TYPE "RUN" TO RE-START.":ON ERROR GOTO 0
0990 KILL N$+".DAT":GOTO 20
1000 END
```

```
 NAM LIST089
*
*            THIS PROGRAM IS DESIGNED TO LIST DS9 TEXT FILE WHICH HAS BEEN
*            COPIED TO A NEWLY INITIALISED DISK, SINGLE SIDE, SINGLE DENSITY.
*
*            IN ORDER TO USE IT, YOU MUST HAVE COPIED AN DS9 TEXT FILE TO
*            A NEWLY FORMATTED DISK WHICH DOES NOT HAVE ANY ERROR SECTORS.
*            THIS IS REQUIRED BECAUSE THIS PROGRAM EXPECTS THE FILE TO BE
*            MADE UP OF CONTIGUOUS SECTORS.
*            USE THE DS9 FORMAT PROGRAM TO FORMAT THE DISK.
*            AS LISTED THE PROGRAM IS SET UP FOR SINGLE SIDE, SINGLE DENSITY,
*            IE 16 SECTORS PER TRACK.
*
*            COMPILE THE PROGRAM TO 'LIST089.CMD',
*            TO USE, ENTER LIST089 (DRIVE-NUMBER)
*            (DEFAULT DRIVE IS 0)
*
*
GETBLK EQU $CD42
STAT EQU $CD4E
PCRLF EQU $CD24
OUTDEC EQU $CD39
PSTRING EQU $CD1E
GETFIL EQU $CD20
WARMS EQU $CD03
GETCHR EQU $CD15
PUTCHR EQU $CD18
OUTCH2 EQU $CD12
INBUF EQU $CD1B
MXTCH EQU 26 27
SETEXT EQU $CD31
RPTERR EQU $CD3F
FMS EQU $D406
DREAD EQU $DE00
DREST EQU $DE09
DSEL  EQU $DE0C
*
*
CRLF  FCC $D,$A,$4
*
INFILE FCB 0,0,0,0
INSEC  RMB 256
INTRACK FCB 1,1 START TRACK,SECTOR
MES FCC $D,$A,/15 DS9 FORMAT DISK IN DRIVE #/
MESDR FCB D,$20,0,0,4
PLINE RMB 256
PLPOS FDB PLINE
*
*
*
START JSR GETBLK
      TFR X,D
      STB INFILE+3
      ORB #$30 MAKE ASCII
      STB MESDR
      LDX 1MES
      LBSR PTERM
WAIT  JSR STAT
      BEQ WAIT WAIT TILL KEY PRESSED
      JSR PCRLF
      LDX #INFIL
      JSR DSEL
      JSR DREST
READ  LDX #INFILE
      JSR DSEL
      LDD INTRACK
      LDX #INSEC
      JSR DREAD
      LDD INTRACK
      INCB BUMP SECTOR
      CMPB #15 <------- CHANGE MAX SECTOR NUMBER FOR OTHER FORMATS *******
      BLS READ01
      LDB #0 RESET SECTOR
      INCA BUMP TRACK
READ01 STD INTRACK
       LDY #INSEC
       LDU 0,Y
       LDX PLPOS
       CMPU #$E5E5
       BNE PRINT
       LBRA WARMS

PRINT  STX PLPOS
       CMPX #INSEC+256
       BEQ READ
       LDA ,Y+
       CMPA #$D END?
       BEQ PRINT1
       STA ,X+
       BRA PRINT

PRINT1 LDA $4
       STA ,X
       LDX #PLINE
       LBSR PSTRING
       LDX #PLINE
       BRA PRINT

PTERM  LDA ,X+
       CMPA $4
       BEQ PTERMX
       LBSR OUTCH2
       BRA PTERM
PTERMX  RTS

       END START
```

'68' Micro Journal
5900 Cassandra Smith Road
Hixson, TN 37343

Dear Mr. Williams:

Leo E. Taylor of West Haven, Connecticut has discovered an incompatibility between TSC Disk BASIC and the PIOSIR patch that Art Weller and I had published in the July 1982 issue. This patch was supposed to fix the problem that Pete Stark described in the April issue and it does so for most applications. It seems though, that it causes TSC BASIC to report a "disk full" error when using virtual arrays and (probably) random files.

In giving this apparent bug in FLEX a little thought, I come to question whether it is really a bug at all. Perhaps the real fault lies in our application programs by their not using the FMSCLS routine at the proper time to re-initialize the FMS and thereby allowing us to change disks. Perhaps this is what TSC had in mind when they developed FLEX.

Anyway, we have a better patch in the works that I hope will be compatible with random files under BASIC. I'm sorry for any inconvience this may have caused any of your readers.

Sincerely,

Randy Kron

Randy Kron
Rt. 2
Ralona, Iowa 52247

William Hartmann
RR 3 Box 131-1
Blue Springs, MO 64015

October 4, 1982

Dear Sir,

I have received several inquires about obtaining a copy of my program DISKFILE (August, 82 MicroJournal). Please advise your readers that this program is available from MicroJournal as disk #7 for a nominal charge. The ad runs in each issue and is on page 47 of the October issue.

Also I have a minor complaint with some of the software that I have bought. Some of the authors assume that the program is going to be run on drives 0 or 1. I have an unusual mix of drives and often run the programs on drives 3 or 4.

I would recommend that the authors use the FLEX system or work drives as appropriate. If this is not practicable then please document the locations that need to be changed to change drive numbers. This is especially true for those programs that are supplies only as binary files.

An associate of my, Neal Bernbner, has found an easy way of defaulting to the FLEX system drive that is not in the "FLEX Advanced Programmers' Guide."

> Set $CC0D (system scratch) to non-zero for default to the system drive or set it to zero to default to the work drive. Then call GETFIL ($CD3D) to set up the parse the line and set up the file specification in the FCB. If a drive number was specified it will be used, else it will default to the FLEX work or system drive as elected by $CC0D.

Yours truly,

William Hartmann

825 N. Sherry Avenue
Norman, Oklahoma 73069
24 October 1982

Don Williams Sr.
68 Micro Journal
P.O. Box 849
Hixson, Tennessee 37343

Dear Don Williams Sr.,

I was pleased to see the program I sent you a year ago published in the October issue of 68 Micro Journal. The program produces an index of all files in a floppy disk library. Unfortunately, the article that explains the program and gives information about the necessary support programs was not published. Although the program contains comments, I am sure many readers do not have the time to trace through the program. A reader might easily miss that the TSC SORT package is required for this program to run. Once running, the prompts from the program provide sufficient instructions to the operator.

To assist your readers, I have placed the source code for the program, the file SORTSPEC.BIN (required), and the text of the article on FLEXNET. As you already know, FLEXNET is an easy to use bulletin board system. It is active 24 hours at (405) 722-6809 and requires the user to have a 300 Baud modem. Please inform your readers that the English explaination of the program and the program itself (to avoid personal keying) are available. The program and article are also available on minifloppy disk for $5 directly from me.

Sincerely,

Thomas J. Weaver

Britt Monk, COP
P. O. Box 602
Elyria, OH 44036
October 20, 1982

68 Micro Journal
P. O. B. 849
5900 Cassandra Smith Rd.
Hixson, TN 37343

Mr. Don Williams, Sr.

The recently released computer game for the TRS-80 Color Computer - "3D Brickaway" - has been sold to the Avalon Hill Game Company.

This company plans to re-release an improved version of the game, not only for the Color Computer, but also for other personal/home computers and video computer systems. Remember, you saw it first on the TRS-80 Color Computer!

Thanks,

Britt

Britt Monk, COP

HELP

I have some information that your readers may be interested in. I have some locations in FLEX to change to allow faster seek and restore times on disk accesses. I am also currently trying to convert SWTPC flex to run with mixed drives and a video board system instead of a terminal system. If anyone has any information that would be of help it would be greatly appreciated. Rick Flick 904 Ginny Ave., 40 Bellevue, NE 68005 (402)291-7538.

CLASSIFIED ADVERTISING

FOR SALE: 6800 CPU $40, 2 16K memory cards $100 ea., 4 4K cards $25, 1 DC2 disk controller $75, 1 PERCOM disk interface $75, 1 THOMAS video board $50. Rick Flick, 904 Ginny Ave 40, Bellevue, NE 68005, (402)291-7538 anytime.

THE COUNT'S CASTLE ADVENTURE, Written in TSC Basic and 18K, $5 for 14 page listing and documentation. $8 on flex mini floppy. (SS,SD) $1 for shipping. Leon Barker, 3611 North 800 West, Ogden, Utah 84404.

SWTPC SYSTEM: MP-A w/SWTBUG;4k,8k & 16k mem bds;MP-C; MP-L; AC-30 cass int; 3 versions BASIC; DC-1; Ramsay Video Bd; Keyboard. Extras: MP-A2; MIKBUG; Buss Ext Bd. Want:$500 or VT-52 to TRS80C. Phone Al, (513)631-0162, 7-9 p.m.

SWTPC 6800/2 with 32K RAM, 8K BASIC version 2.3, CT-64 terminal CT-VM video monitor, AC-30 cassette interface, $1000. Richard Price, 73090 McKay, Romeo, MI 48065.

SWTPC 6800, 32K Newteck Music board, MPR, 3 MPS boards. PERCOM disk drive with controller,CT-64. Many extras. Best offer over $800. Mihran Kochyan, (313)271-3594.

WANTED: Miniflex users to band together for mutual assistance. Let's support our DOS. Preston Brashear, 1580 Eastgate Drive 320, Garland, Tx 75041, (214)270-0053 271-7783.

FOR SALE: Used HELIX Computer System 50% off. Includes HELIX 6809 CPU, 64K RAM. Call (314)291-2728.

WANTED OR FOR SALE: We need two identical terminals. We have one CT-82. Buy ours for $600 and I will get two of something else. OR sell me yours and I will have what I need. (207)276-5350 day or (207) 244-7444 night, Howard Johnson, Mt Desert Island High School, Mt Desert, Me 04660.

FLEXPLUS DOS

Advantages of FLEXPLUS are:

- Best price anywhere
- Easy start-up — just type "RUN FLEX+"
- Allows you to save RS compatible disk files from FLEX-PLUS
- All FLEX compatible software will run including INTERRUPT DRIVEN SOFTWARE
- NO HARDWARE MODIFICATIONS NEEDED
- Warranty will not be voided — no need to open computer
- Wide range of available software
- Requires Supercharger board

FLEXPLUS DOS

FLEXPLUS is a powerful, easy-to-use disk operating system. Spectral Associates has adapted TSC's FLEX to the best DOS completely compatible with Radio Shack software for use on the Color Computer. Eliminate the need for Radio Shack's TRS DOS. Use FLEXPLUS with Editor/Assembler and have the options of a full range of utilities. FLEXPLUS works on the 32K Radio Shack disk system with 64K memory chips with a High Resolution multi-screen format that supports a 24 line by 51 character display! Also included are special enhancements to Radio Shack's Disk system when you are running FLEX with single or double sided, single or double density, 35, 40 and 80 track drives.

SUPERCHARGER

If you have a 32K E series printed circuit board, the SUPERCHARGER will let you switch your computer to full 64K RAM mode. It just plugs into the ROM PAK port and you can use BASIC or not as you wish. NO MODIFICATION is needed and it will not void your warranty. It allows you to still plug ROM pack and/or the disk controller board into the computer. ONLY $39.95

FLEXPLUS COMPATIBLE SOFTWARE

SLEUTH DISASSEMBLER — A program which allows user to examine, modify and disassemble object program files on disk or in memory on 6809 systems. Will convert an object file in RAM to a text file on disk.
Catalog No: UT-221 99.00

TALBULARASA (Electronic spread sheet) — These programs enable the user to generate and maintain tabular computation schemes, providing a simple user interface and sophisticated report generation.
Catalog No: UT-222 100.00

CROSS ASSEMBLER — A set of modules for 6800/1, 6805, 6802, 8080/5, and Z-80 micro processors which will enable the user to convert a master assembly module to assemble code for any of the above micros.
Catalog No: UT-223 50.00 ea. or 3/100.00

LUCIDATA PASCAL — A well-established, user friendly implementation of the popular high level language Pascal. Specifications include 5 constants, pre-declared types, simple data types, and data structures (ARRAY, STRING, and RECORDS). Special features for the advanced user is a provision of many EXTERNAL functions e.g. for obtaining the absolute address of variables; the stack/heap margin; performing unsigned addition and subtraction for address calculation; block read/write for high speed transfer of blocks of memory; linkage to user-written assembler code.
Catalog No: UT-232 205.00

TSC 6809 BASIC (6-digit math) Catalog No: UT-227 75.00

TSC 6809 Advanced BASIC (17-digit math)
Catalog No: UT-228 100.00

BASIC Precompiler Catalog No: UT-229 50.00

Advanced BASIC Precompiler Catalog No: UT-230 50.00

TSC PASCAL
Catalog No: UT-231 200.00

All of the above Flexplus compatible software comes on disk.

FLEXPLUS comes complete with Supercharger, MACRO Editor/Assembler with conditional assembly and 160 page manual. All for $149.95.

LOOK AT THIS!

Expand Memory from 16K to 32K.

RAMCHARGER - for series D and earlier printed circuit board.

DYNOCHARGER - for series E printed circuit board.

Either for only $79.95.

SPECTRAL ASSOCIATES

139 Harvard Ave.
Tacoma, Washington 98466
(206) 565-8483

WRITE FOR COMPLETE CATALOG
ADD 1% FOR SHIPPING NO COD
VISA OR MASTERCHARGE ACCEPTED
All prices in U.S. funds

16K MACHINE LANGUAGE GAMES

- GHOST GOBBLER $21.95
- DEFENSE $21.95
  For Missile Command enthusiasts.
- SPACE WAR $21.95
- GALAX ATTAX $21.95

- PLANET DEFENSE $21.95
  Like the arcade version of DEFENDER.
- CROID $12.95
  Carry on an intelligent conversation with the computer with this version of the ELIZA artificial intelligence program.
- CC THELLO $14.95
  Super arcade quality OTHELLO game.
- KEYS OF THE WIZARD $19.95
  Best new adventure game! Over 200 rooms filled with creatures, tricks, treasures and magic spells. Cassette save feature built in.
- COLOR OUT $9.95
  Like Breakout, you must knock out six layers of blocks using your paddle to prevent missiles from escaping. Joysticks optional.

ORDERS ONLY TOLL FREE

800-426-1830

GHOST GOBBLER

UTILITIES

- EDITOR/ASSEMBLER $34.95
- SUPER MONITOR 19.95
- EPROM PROGRAMMER 89.95
  (Program your own ROMs for the ROM PAK port)
- MAGIC BOX 24.95
  (Load MOD III Basic Tapes into your computer)

THE FACTS

At last a complete description of the "guts" of the Color Computer. Specs on all of IC's complete schematics, theory of operation and programming examples.

$14.95

TRS 80 **COLOR COMPUTER** TRS 80

TEN MOST-ASKED QUESTIONS

ABOUT DYNACALC™

THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

1. **What is an electronic spread-sheet, anyway?**
   Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.

2. **Is DYNACALC just for accountants, then?**
   Not at all. DYNACALC can be used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. There's even a built-in sort command, so you could use DYNACALC to manage small data bases - up to 256 records.

3. **What will DYNACALC do for ME?**
   That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalc, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications.

4. **Do I have to learn computer programming?**
   NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. in Computer Science can understand it. Built-in HELP messages are provided for quick reference to operating instructions.

5. **Do I have to modify my system to use DYNACALC?**
   Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.

6. **Will DYNACALC read my existing data files?**
   You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, RMS data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.

7. **How fast is DYNACALC?**
   Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.

8. **Is there a version of DYNACALC for MY system?**
   Probably. You need a 6809 computer (32k minimum) with FLEX or UniFLEX operating system. A version for OS-9 is also in the works. You also need a decent CRT terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UniFLEX version of DYNACALC also allows you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX and Data-Comp's F-MATE. A version for Frank Hogg's Color Computer FLEX is also being done.

9. **How much does DYNACALC cost?**
   The FLEX versions are just $200 per copy; UniFLEX version $395. Foreign orders add $10 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.

10. **Where do I order DYNACALC?**
    See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 a.m. to 6 p.m., Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX versions. Software serial number is required for the UniFLEX version of DYNACALC.

ORDER YOUR DYNACALC™ TODAY

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 7A Burton St., Darlinghurst, NSW 2010 Sydney. Telephone: 02-357-5111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center

13461 Olive Blvd.

Chesterfield, MO 63017

(314) 576-5020

UniFLEX software prices include maintenance for the first year.

DYNACALC, DYNAMITE, and DYNAMITE + are trademarks of Computer Systems Center.

F-MATE is a trademark of Data-Comp.
VisiCalc is a trademark of VisiCorp.
OS-9 is a trademark of Microware and Motorola.
FLEX and UniFLEX are trademarks of TSC.

ALSO FROM CSC

DYNAMITE +

"THE CODE BUSTER"

now available for UniFLEX OS-9 version soon

DYNAMITE + is a new version of DYNAMITE, our popular 6809/6800 disassembler package for 6809 FLEX. Present users of DYNAMITE can upgrade to DYNAMITE + by sending us the original DYNAMITE diskette and $40 (plus $5 for foreign postage). DYNAMITE + does everything DYNAMITE does, and more! A cross-reference generator has been added, label files are now maintained only in text form (LABEL EQU $xxxx), and boundary file specifications have been tremendously simplified, which makes it easier to disassemble large programs containing lots of big tables.

The UniFLEX version of DYNAMITE + does everything the FLEX version does, and also automatically handles system calls and 'Info' areas.

DYNAMITE + is available for $100 per copy on FLEX (specify diskette size), and $300 on UniFLEX. Foreign orders add $5 per copy for postage.

DATA SYSTEMS 68 DATA SYSTEMS 68

We want YOU to buy your boards from us and see the difference for yourself.

8" DOUBLE DENSITY DISK CONTROLLER

- Double sided, single/double density
- Write-precompensation
- Disk drivers included
- Requires oscilloscope for setup
- Phase lock loop data separator
- Uses the Western Digital 1791 controller chip
- Requires DMA INTERFACE BOARD for double density

$39.50 bare board

$225.00 assembled and tested

5¼" DOUBLE DENSITY DISK CONTROLLER

- Double sided, single/double density
- Write-precompensation
- Regulators to power 2 disk drives
- Disk drivers included
- Phase lock loop data separator
- Uses the Western Digital 1791 controller chip
- Requires oscilloscope and voltmeter for setup

$39.50 bare board

$225.00 assembled and tested

6809 CPU BOARD

- On-board Baud Rate Generator using the MC14411
- Supports extended addressing
- Space for four 2716 Eproms allowing you to switch between sets
- Solder masked top and bottom
- Component placement silk screened on top of board

$32.50

MOTHER BOARD

- .093 board no flex
- 8 50-pin slots 8 30-pin slots
- 4 or 16 addresses per slot
- fully decoded
- I/O configuration by way of 1 16-pin header and 4-pos dip switch
- Baud Rate Generator is on board 9600, 4800, 600, 300

$62.50

MODEM BOARD

- Uses the MC6860 Modem Chip
- Discrete active filters
- Works with a CBT Data Coupler
- Bell 103 compatible
- Originate and auto answer
- With the MC6850 it looks like a serial port to software
- No hard-to-get parts (30-pin)

$19.50

DMA INTERFACE BOARD

Ugrade your DS 68 Double Density Disk Controller to run full DMA on the 6809

$29.50 bare board

6845 VIDEO DISPLAY BOARD

- Utilizes the 6845 CRT Controller
- Software selectable format up to 80 by 24
- Character font in one 2716 Eprom
- 2K Screen Buffer
- Selectable on any odd 2K boundary

$32.50

6847 VIDEO GRAPHICS BOARD

- Uses the MC6847 Video Display Generator
- MC1372 RF Modulator on board
- All VDG modes selectable by way of an 8-position Dip-Switch
- Addressable on any 8K boundary using 2114 RAMS in 8K blocks

$32.50

30 Pin & 50 Pin Extender Boards

All Extender boards fully labeled and shielded to prevent RF Interference.

$19.50 EACH

MULTIPLE I/O BOARD

- 2-ACIA's and 2-PIA's
- Regulated +12, −12, and +5 available for key board
- Addressable from $E7EX-$E7FX
- Addressed to overlay Motherboard I/O
- Buffered handshake on Serial I/O

$32.50

64K DYNAMIC RAM BOARD

- Completely transparent refresh (during 01) at 1 MHz
- Operates with both 6800 and 6809 systems
- Compatible with the 20-bit extended addressing mode
- Low power − 12V at 150 ma., 5V at 500 ma., and −5V at 7 ma.
- Uses 4116-type RAM with 200 NS access time
- Designed for Motorola MC3242A and MC3480 Dynamic Memory Control Chips
- No timing problems
- No one-shot delays
- No adjustments

$39.50

DUAL SERIAL BOARD

- Combines two ACIA's at the same port
- An optional NMI debouncing circuitry is on board
- All line are RS-232 levels (30-pin)

$19.50

- All Boards Solder Masked Both Sides
- All Silk Screened Nomenclature
- Full Documentation Included

- VISA & Master Card Accepted
- Add $3.00 for shipping & handling
- Add $1.00 for C.O.D.

Data Systems "68"
2316 Diversified Way
Orlando, Florida 32804

(305) 425-6800

Data Systems "68"
2316 Diversified Way
Orlando, Florida 32804

TERMS: Check, MasterCard, Visa. C.O.D.'s. Florida residents add 5% sales tax. Specify type board(s) and quantity of each when ordering.

Please Note: DATA SYSTEMS 68 is not associated with any magazine or any other computer company.

OS/9, FLEX, COLOR FLEX, UNIFLEX Software*

SUPER SLEUTH DISASSEMBLER $99-FLEX $100-UNIFLEX $101-OS/9
This program processes 6800/1/2/3/5/8/9/6502 programs, enabling the user to analyze, modify, and disassemble (with labels) object code, with output to terminal, printer, and disk, and cross-reference and label-definition capabilities.

Z-80/8080/5 SUPER SLEUTH DISASSEMBLER $99-FLEX $100-UNIFLEX $101-OS/9
This version of SUPER SLEUTH processes Z-80/8080/5 object code on the 6800/1/9.

CROSS-ASSEMBLERS each $50 3/$100-FLEX each $60 5/$120-UNIFLEX
These programs and TSC macros enable the user to process 6800/1, 6805, 6502, Z-80, 8080/5 programs in original format.

6805 and 6502 DEBUGGING SIMULATORS each $75-FLEX $80-UNIFLEX $100-OS/9
These programs enable the user to interactively analyze, modify, and debug (14)6805 and 6502 object code.

6502-TO-6809 XLATOR SYSTEM $75-FLEX $80-UNIFLEX $85-OS/9
This program enables the user to translate 6502 assembler code into 6809 assembler code, noting inexact conversions.

6800-6809 & 6809 PIC XLATORS both $50-FLEX $60-UNIFLEX $75-OS/9
These programs enable the user to translate 6800/1 assembler programs to 6809 mnemonics and to convert 6809 programs to position-independent code and data, using PC, S, U, X, and Y as base registers.

UNIFLEX SIMULATOR FOR FLEX $100-FLEX $110-UNIFLEX
This program enables the user to debug UNIFLEX assembler programs using the TSC DEBUG and other facilities of FLEX.

OS/9 SIMULATOR FOR FLEX $101-FLEX
This program enables the user to debug OS/9 assembler programs using the TSC DEBUG and other facilities of FLEX.

FULL SCREEN FORMS DISPLAY (6809 X-BASIC) $50-FLEX $75-UNIFLEX
These programs enable the user to define and generate table-driven full-screen display and data-entry programs.

FULL SCREEN MAILING LIST (6809 X-BASIC) $100-FLEX $110-UNIFLEX
These programs enable the user to define and maintain mailing-list-oriented data bases.

FULL SCREEN INVENTORY/MRP (6809 X-BASIC) $100-FLEX $150 UNIFLEX
These programs enable the user to define and maintain inventories, and include hierarchical materials requirement planning.

TABULA RASA SPREADSHEET (6809 X-BASIC) $100-FLEX $200-UNIFLEX
These programs enable the user to generate and maintain tabular computation schemas, providing a simple user interface and sophisticated report-generation, similar to DESKTOP/PLAN (TM Desktop Computing).

TSC BASIC/XPC UTILITY PROGRAMS all $25-FLEX $50-UNIFLEX
These programs enable the user to resequence or cross-reference any Basic program and generate XPC Basic sort programs.

For VISA and MASTER CARD give account, exp date, phone
U S funds only - add 5% (10% foreign) for shipping

Computer Systems Consultants, Inc.
1454 Latta Lane, Conyers, GA 30207
Telephone Number 404-483-1717/4570

---

68 MICRO JOURNAL PROGRAMS on DISK

Disk #1: FILESORT, MINICAT, MINICOPY, MINIFMS, **LIFETIME, **POETRY, **FOODLIST, **DIET.
Disk #2: DISKEDIT w/ inst. & fixes, PRIME, *PRMOD, **SNOOPY, **FOOTBALL, **HEXPAWN, **LIFETIME.
Disk #3: CBUG09, SEC1, SEC2, FIND, TABLE2, INTEXT, DISK-EXP, *DISKSAVE.
Disk #4: MAILING PROGRAM, *FINDDAT, *CHANGE, *TESTDISK.
Disk #5: *DISKFIX 1, *DISKFIX 2, **LETTER, **LOVESIGN, **BLACKJAK, **BOWLING.
Disk #6: **PURCHASE ORDER, INDEX (Disk file indx).
Disk #7: Linking Loader & RLOAD, Harkness
Disk #8: CRTSET, Lanpher (May '82)
Disk #9: DATECOPY, DISKFIX9 (Aug '82)

NOTE: All are as published or received by 68 Micro Journal, some have fixes and patches.

This is a reader service only! No Warranty is offered or implied, they are as received and are for reader convenience ONLY. Also 6800 and 6809 programs are mixed, as each is fairly simple (mostly) to convert to the other.

PRICE: 8" Disk $19.95 - 5" Disk $17.95

68 MICRO JOURNAL
POB 794
Hixson, TN 37343
615-842-4600

* Indicates 6800, ** Indicates BASIC SWTPC or TSC - 6809 no Indicator.

MASTER CARD - VISA accepted - Foreign add sufficient postage surface or air!!

NEW 6809 SYSTEM!

Now, for about the same price as you would expect to pay for the memory capacity alone, you can have a complete single board computer with these features:

- 6809 CPU, 1MHz clock
- 192KB RAM included, sockets for 64KB more
- 84X24 display of a 7X12 character font
- Keyboard interface for an un-encoded switch matrix
- Floppy controller for two 5" drives, single or double sided
- Parallel printer port
- Serial I/O port
- General purpose 8-bit parallel I/O port
- Parallel expansion port
- Dimensions: 8.6 by 10.3 inches

NEW FEATURE: Pseudo-disk with 112KB capacity. Now you can use RAM to increase the performance of programs that use disk files.
Assembly programming is optional and is not required for system set-up.

The FLEX operating system is supported by our device drivers. BASIC, PASCAL, and C are available for FLEX. The device drivers (in EPROM) include advanced features like auto-repeat for the keyboard, and track buffering for the disks. Commented source code of all EPROM contents is supplied.

For more information, send a stamped self-addressed envelope and we will send you a configuration guide that explains how to set-up a system. An assembled board is purchased by sending check or money order for $735 per board. (California add 6% sales tax)

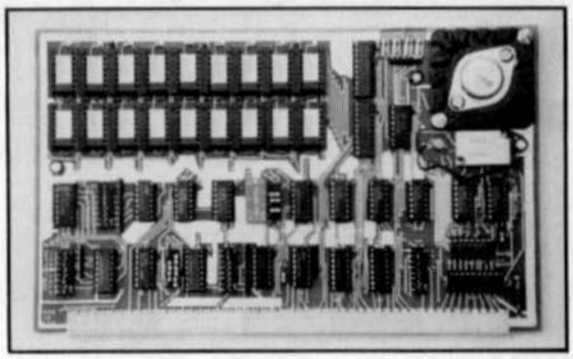
Chandler Microsystems
22051 COSALA
MISSION VIEJO, CA 92691

FLEX, trademark Technical Systems Consultants, Inc.

WINCHESTER FOR MOTOROLA EXORCISOR/MDOS

☐ 10 MB Winchester hard disk runs MDOS on Motorola Exorcisor System ☐ No modification to MDOS required ☐ MDOS based software stays alive ☐ All user software operates without modification ☐ Optional SA-801R flexible diskette drive system ☐ Optional 10 MB removable cartridge.

CSA *For information call (714) 566-3911*
Computer System Associates
7562 Trade Street, San Diego, CA 92121



QMM1-B 256K MEMORY FOR SS50-C 6809 SYSTEMS

Compatible with systems by **SSB, GIMIX,** and **SWTPC** including those with **DMA** disk controllers.

Full 2Mhz operation with transparent on board refresh, runs continuously at 2 Mhz without cycle stretching or stealing.

Very versatile addressing and disable features.

Parity option halts processor and sounds audible alarm upon detecting a read error.

All boards assembled, tested, burned in and warranted for 1 year.

Also available with 64K, 128K, or 192K.

256K for $1135.00 — 256K w/ parity $1270.00

Delivery: Stock—2 weeks. Terms: Prepaid or COD.

D.P. Johnson (503) 244-6152
7655 S.W. Cedarcrest St., Portland, OR 97223

OS9 Application Software

Specialty Electronics, Inc.

ACCOUNTS PAYABLE

Take the "headache" out of tracking your accounts payable with Specialty Electronics Interactive Accounting System. The accounts payable package supports these outstanding features:

1. Entry of debits, credits, regular invoices, full and partial payments.
2. Hand checks printed directly or computer printed checks with stubs and check ledger.
3. Provides aging of accounts and proper general ledger distribution of all entries.
4. Job costing, customer order number tracking and buyer identification are provided.
5. Vendors may be added as needed.
6. Complete audit trails are provided.
7. Reports can be generated for specific due dates, customer groups, open or closed items and in either voucher detail or vendor summary formats.

Accounts Payable I-code
$299

GENERAL LEDGER with CASH JOURNAL

The general ledger is the center of the Specialty Electronics Accounting System. With the package you can:

1. Print up balance sheets and income statements in various formats.
2. Define account names, spacing, positioning, headings and [illegible].
3. Format special reports and print percentages.
4. Post by hand cash journal or by using the interactive accounts receivable, payable and payroll.
5. Provide a clear audit trail for all entries.
6. Input data in an easy to follow format.
7. Use for multi company accounting without modification.

General ledger I-code
$399

ACCOUNTS RECEIVABLE

Your Accounts Receivable can be followed with a minimum of time investment using these features:

1. Regular invoicing, debit and credit memos, full and partial payments.
2. Progressive billing and payments.
3. Aging of periods specified by the user.
4. New customers entered as needed.
5. Statements are generated listing individual invoices and overdue amounts, totaled by aging category.
6. Total interaction with the general ledger with tax, shipping and travel expenses computed, keep daily and posted to various accounts.

Accounts Receivable I-code
$299

INVENTORY

The Specialty Electronics Interactive Accounting System Inventory Control Package provides the tools for complete control of a large and active inventory, including:

1. Reports for quantities on hand, quantities on order activity and many other categories.
2. Complete item description, category groups, supplier information, order dates, reorder quantities, etc.
3. Simple input and reconciliation procedures.

Inventory Control I-code
$299

PAYROLL

The Specialty Electronics Interactive Accounting System provides payroll support which goes beyond writing paychecks. Its features include:

1. Weekly, bi-weekly, semi-monthly and monthly pay periods.
2. Hourly, salary, vacation, holiday, commission, overtime and compensatory pay categories and tips.
3. Deducts federal and state payroll taxes, insurance, additional tax or special deductions.
4. Daily time keeping allowed.
5. Prints checks, stubs, check ledger and journal history.
6. Prints W-2 forms, federal and state tax report information.
7. Keeps full employee history.
8. Tax tables allow user modification.

Payroll I-code
$425

Complete Documentation $19.95

' OS9 and Basic 09 are trademarks of Microware, Inc. and Motorola Corp.

P.O. Box 541
2110 W. Willow

Specialty Electronics

(405) 233-1632
Enid, OK 73701

'68' MICRO JOURNAL

★ The only ALL 6800 Computer Magazine.

★ More 6800 material than all the others combined:

MAGAZINE COMPARISON
(2 years)
Monthly Averages
6800 Articles

KB	BYTE	CC	DOBB'S	TOTAL PAGES
7.8	6.4	2.7	2.2	19.1 ea. mo.

Average cost for all four each month: $6.53
(Based on advertised 1-year subscription price)

'68' cost per month: $2.04

That's Right! Much, Much More
for About
1/3 the Cost!

OK, PLEASE ENTER MY SUBSCRIPTION

Bill My: Master Charge ☐ — VISA ☐

Card #__________ Exp. Date__________

For ☐ 1-Year ☐ 2 Years ☐ 3 Years

Enclosed: $__________

Name____________________

Street____________________

City__________ State________ Zip______

My Computer Is:____________________

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

SUBSCRIPTION RATES

USA
1 Year $24.50, 2 Year $42.50, 3 Year $64.50

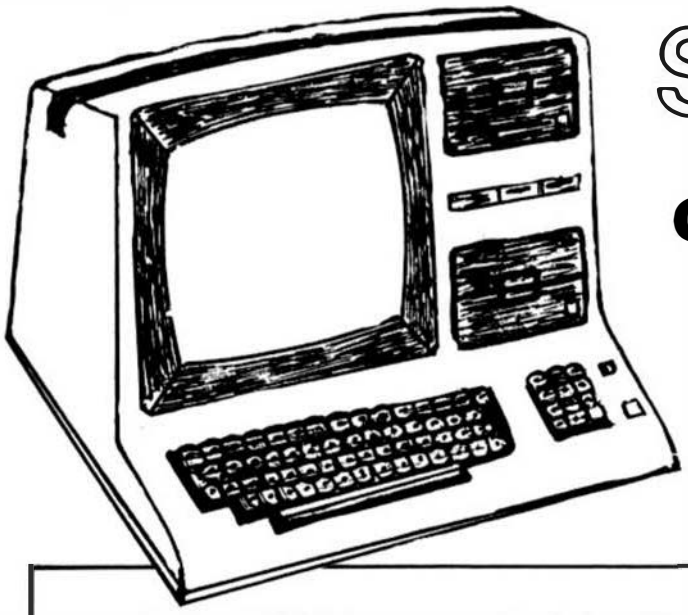
*FOREIGN SURFACE Add $12.00 per Year to USA Price

*FOREIGN AIRMAIL Add $36.00 per Year to USA Price

**CANADA & MEXICO Add $5.50 per Year to USA Price

Cash (USA) or drawn on a USA Bank!!!

Universal Data Research, Inc. Introduces



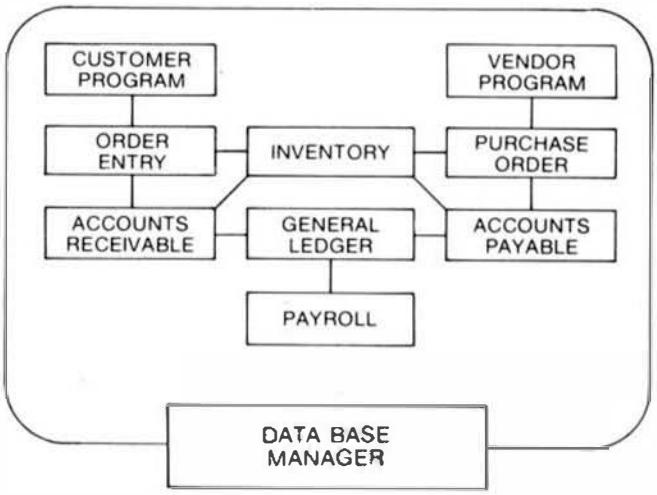
SOFTWARE
for the COLOR COMPUTER and the TRS 80 Model III

- Data Base Manager _____ $150
- Church Contributions Package _ $150
- Single Entry Ledger _____ $ 95

FLEX* and UniFLEX* SOFTWARE for the 68XX Operating System . . .

INTEGRATED BUSINESS PROGRAMS
for large and small companies

	FLEX	UniFLEX
Accounts Receivable	$395	$495
Accounts Payable	$395	$495
General Ledger	$395	$495
Inventory 2	$395	$495
Payroll	$395	$495
Data Base Manager	$450	$550



STAND-ALONE SOFTWARE

- Single Entry Ledger
- Check Balancer
- Costing and Estimating
- Church Contribution Pkg.
- Church School Billing
- Fund Raising
- Custom Programming
- M.F.I. (Extensive Mfg./P.O.)
- Bulk Mailings
- Maintenance Schedule
- Vendor Program
- Purchase Order
- Accounts Payable
- Accounts Receivable
- Customer Program
- Order Entry

UNIVERSAL DATA RESEARCH INC.

*FLEX & UniFLEX are Trademarks of Technical Systems

2457 Wehrle Drive,D-1, Buffalo, New York 14221
Phone (716) 631-3011

master charge VISA

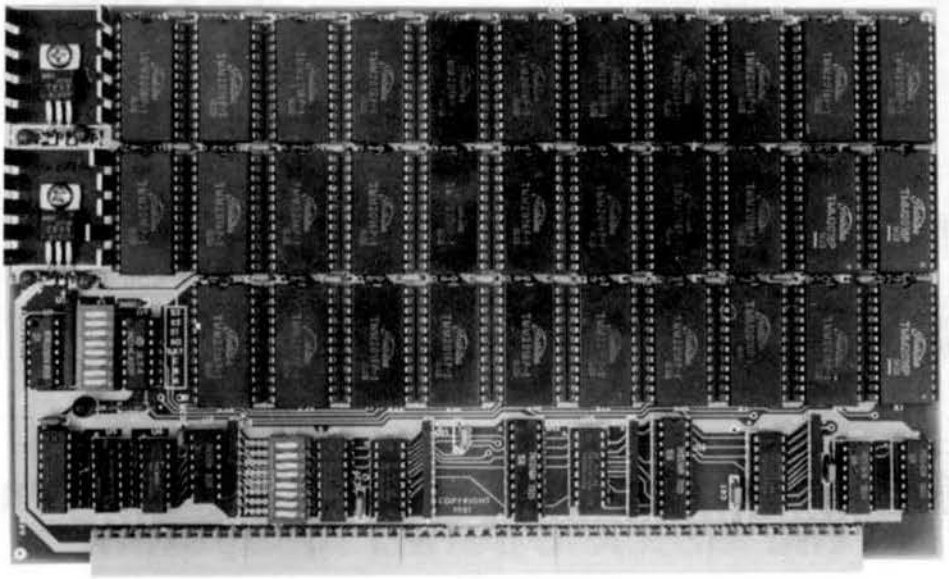
64K SS-50 STATIC RAM

NEW!

$219.00 (48K KIT)

NEW!

LOW POWER!



RAM OR EPROM!

BLANK PC BOARD WITH DOCUMENTATION $52

SUPPORT ICs + CAPS -	$18.00
FULL SOCKET SET -	$15.00

ASSEMBLED AND TESTED **ADD $40**

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K Kit	$269
64K Kit	$319

16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Order under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50. add 85¢ for insurance.

ARCADE 50

POWERFUL COLOR GRAPHICS
Uses the new TMS9918A Video Display Processor. High resolution 256 x 192 pixel display with 15 colors. 16k Bytes of onboard RAM does not reduce user memory. 32 graphic images can be individually moved with simple X-Y commands for smooth animation.
External video input allows subtitling.
NTSC composite video output.
SOUND EFFECTS AND MUSIC
Three AY3-8910 Programmable Sound Generators.
Nine simultaneous voices.
Three independent noise sources.
Onboard stereo amplifier drives two 8 ohm speakers.
ADDITIONAL I/O CAPABILITIES
Eight analog inputs with 8 bit resolution.
Supports four joysticks with pushbutton switches.
Eight bit parallel I/O port.
Entire unit maps into 256 bytes of memory.
DOCUMENTATION AND SOFTWARE
Programming manuals for Video and Sound Processors.
Subroutine library and Super Demo Maze Game.
Example programs in BASIC, FBASIC and ASSEMBLY.
User library and sales support.

ARCADE 50 assembled and tested	$325.00
Video and Audio connector set	15.00
4 Joystick connector set	15.00
2 Radio Shack Joysticks	24.00
UHF channel 33 modulator	32.00
Gold Molex connectors	12.00
A/BASIC for 6800	110.00
FBASIC for 6809	110.00
FBASIC (with ARCADE 50)	75.00
FBASIC (manual only)	10.00
ARCADE 80 (TRS Model I)	395.00
ARCADE 100 (S-100 BUSS)	375.00
ARCADE 50 RGB	375.00
LABVIDEO (Motorola EXORbus)	375.00
LABVIDEO RGB	375.00
NEW MV09 6809 Processor Board	225.00
*Comes assembled with PIA and ACIA	
*12 Sockets for 2716, 2732 or RAM	
*Supports DMA disk I/O	
*Ideal for 6809 upgrade or process control	
AMDEK COLOR I Monitor	425.00
AMDEK COLOR II Monitor	799.00
AMDEK COLOR III Monitor	499.00
256K Dynamic Memory Board (assembled)	795.00
256K Dynamic Memory Board (assembled w/64K)	395.00
64K Dynamic Memory Board (assembled)	295.00

Specify 5" or 8" soft sector disk for TSC's FLEX or MICROWARE S OS/9 system.
TERMS: CASH, VISA, MC, C.O.D.

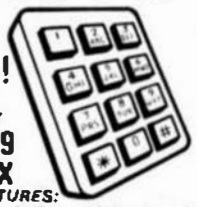
FBASIC

TERMINUS DESIGN INC., in conjunction with Microware Systems Corporation, is proud to announce FBASIC-an enhancement of Microware's 6800 A/BASIC. Their fast compiled BASIC has been adapted for 6809 users with added video and sound features for ARCADE 50 users. FBASIC is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions, Disk I/O and fast integer arithmetic with multiple-precision capability. Graphics verbs and functions fully support the Arcade 50. Arcade statements include:

INIT	MODE	BLANK	BACKDROP
SIZE	MAG	VREG	DELAY
MOVE	DRAW	FCOLOR	JSWITCH
REMOVE	RDRAW	BCOLOR	SWITCH
PSG	TONE	ENVL	VOLUME
ADC	SPRITE	SPNAME	ENDEF
SPCOLOR	RSPRITE	SPDEF	PATDEF
VPEEK	VPOKE	VPRINT	

TERMINUS DESIGN INC
16 SCARBROUGH ROAD
ELLENWOOD, GA 30049
(404) 474-6885

AUTO - COMM



NEW! For 6809 FLEX

The 'modem' program that automates time sharing communications.

FEATURES:
- Automatic dial-up (dialer schematic provided)
- Tree structured "HELP" command system
- Smart command interpreter responds to English commands and to all command abbreviations
- Transmit manually or automatic transmission of sign-on messages
- Allows any number of message & phone number files
- Receives and saves information on disk. Automatically determines if remote computer responds to XON/XOFF, and switches to memory-size-limited mode, if required
- Transmit text disk files of any length to another computer
- Full duplex, half duplex, echo, no-echo, add L.F. to C.R., or no add L.F., 8 UART data formats, slow transmit mode for speed-below-baud-rate systems
- Has prompt character mode for transmitting to an editor (sends a line then waits for editor prompt for new line)
- Delayed or real time printing and/or disk file save
- 25 self documented commands

Price: for 1st 2 months introductory rate - $59.00 (Reg. $75)
Manual only: $3.95
Special reduced prices for computer club multiple purchases

SYSTEMS designware

6712 E. PRESIDO ST. SCOTTSDALE, AZ. 85254
ORDER TOLL FREE 800-272-4817
ARIZONA CALL COLLECT (602) 991-1657
FOR INFORMATION CALL (602) 991-1657

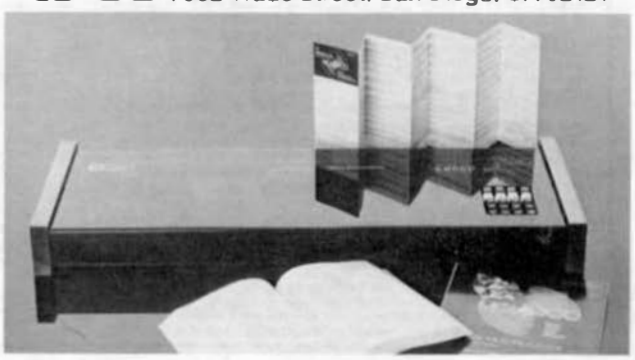
ENGINEERS/TECHNICIANS

THE MICRO 68000 IS DESIGNED FOR YOU!

COMPLETE, READY-TO-GO SYSTEM INCLUDES:

☐ 6 amp switching power supply ☐ Keyboard ☐ Display - Hex & Binary ☐ Pete Bug keyboard monitor ☐ Optional Macs Bug CRT monitor ☐ Attractive cabinet ☐ Dual RS232 interface ☐ 32 bit parallel I/O ☐ Versabus compatibility ☐ The only system that provides for direct entry of 68000 machine code.

CSA *For information call (714) 566-3911*
Computer System Associates
7562 Trade Street, San Diego, CA 92121



SMOOTH™ Software

ALL IN ONE
Editor - Text Processor - Mailing Labels
Mailing Lists - Use any CRT terminal and printer

Supports Editing commands such as bottom, change, delete, find, insert (single line), input (multiple lines), list, next, overlay (with cursor editing, character deletion and insertion), overstrike (for selected darker text), print, restart, set, top, underline, up, and verify.

Supports Text Processing commands such as block copy, block move, centering, margin justification (widen and narrow), paging, and tabbing.

Mailing Lists and Labels. Use the same mailing list disk file (with protected areas) for both mailing labels and repeat letters. Repeat letters are personally adressed to each person or selected persons on the mailing list.

Most Powerful File Handler found in any editor. Append one file to the end of another, or insert (merge) one file into another as designated by the line pointer. Print specified lines to your printer or to a disk file. Edit files larger than the text buffer. Does not produce output files when not desired. Delete disk files from the editor.

Printer commands. Control charactera can be sent to the printer for formal control either directly from the control terminal or by imbedding them in the text. The set command contains interface initialization and character output routines to support the SWTPC MP-C interfaceas well as the standard serial and parallel interfaces. Jumpsare also provided to user supplied printer routines. User selects the portaddress (0 thru 7, A or B) thereby eliminating the need for the user to install printer software routines. Editor can be initialized for either 4 or 16 addresses per port.

Editor allows exiting to either the monitor or DOS and then reenter (Warm Start) without destroying previously prepared text in the buffer. The Restart command erases contents in the buffer without the user having to reload the Editor.

The Editor allows the user to toggle between full duplex (no echo) and half duplex (echo) as needed. It responds to commands in both upper and lower case and can be used to create assembler source code and Basic programs as well as text.

Specify 6800 or 6809, SS8 or FLEX™, 5" or 8"	75.00
Printed source listing is available for an additional	35.00
All-In-One, Write'n Spell, and Spell'n Fix package	195.00

Software by Technical Systems Consultants, Inc.

Flex™ (includes Editor and Assembler)	150.00
UniFLEX™ (includes one year maintenance and update)	550.00
Editor	50.00
Assembler	50.00
68000 Cross Assembler on 6809	250.00
6809 Cross Assembler on 6800	100.00
Text Processor	75.00
Extended Basic	100.00
Basic Precompiler (specify standard or extended)	50.00
Basic for UniFLEX™ (includes one year maintenance and update)	200.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™) (includes one year maintenance and update)	300.00
Sort/Merge Package	75.00
6809 Flex™ Utilities	75.00
6800 Flex™ Utilities	100.00
Debug Package	75.00
Diagnostic Package	75.00

Software by Microware Systems Corp.	UPDATE	SOURCE	MANUAL	OBJECT
OS-9™ Level One Operating System	75.00	00.00	40.00	200.00
OS-9™ Level Two Operating System	75.00	N/A	40.00	500.00
ASIC0 ™	75.00	N/A	25.00	200.00
BASIC0 ™ Run-Time Package				100.00
OS-9™ Macro Text Editor		300.00	15.00	125.00
OS-9™ Interactive Assembler		300.00	10.00	125.00
OS-9™ Interactive Debugger (Disk version)		100.00	10.00	50.00
CIS Cobol Compiler	250.00	N/A	80.00	900.00
CIS Cobol Run-Time Package				400.00
Pascal Compiler	100.00	N/A	0.00	00.00
Pascal Run-Time Package				100.00
Microware yearly support service ($200.00 for OS.9 Level 2)				75.00

SWTPC		Kit Assembled
DMF2C Disk Controller Board		595.00
6809 SWTPC FLEX™ Disk and manual		35.00
6809 SWTPC FLEX™ Disk without manual		15.00
DC-4 Disk Controller	N/A	230.00
SBUG-E (2716 compatible)		25.00
MP-A2 6800 CPU BOARD		150.00
MP-S Serial interface (single port, limited quantity)	80.00	60.00
MP-S2 Serial interface (dual port)	N/A	120.00
MP-LA Parallel interface (dual port, limited quantity)	40.00	60.00
MP-L2 Parallel interface (dual port)	N/A	120.00
MP-R Single voltage 2716 prom programmer	N/A	114.50
MP-N Calculator board	54.95	92.00
MP-T Interrupt timer	N/A	92.00
S32 Universal Static Memory Board	N/A	124.50
MP-09 6809 CPU board (Used $225.00)	N/A	295.00
69A Chassis, P/S, 68B09 CPU, 8K, RAM, One Serial Port	N/A	N/A

Universal 68XX MBT Bare Motherboard. 6800/6809, 4/16 addresses per port.	
8 50 Pin/6 30 pin slots, baud rate generator, 6840 interrupt timer slow device circuitry, 15 1/8" x 9 3/8"	55.00

Connectors (10 pin)	Tin	Gold
Male with square pins	0.50	1.50
Female	0.50	1.50

Special Software	
4K 6809 HUMBUG	75.00
4K 6800 HUMBUG (RAM needed at $A000 and $D000)	65.00
2K 6800 HUMBUG (With cassette LOAD and PUNCH)	40.00
2K 6800 HUMBUG (Extra commands instead of cassette software)	4000
Other HUMBUG versions including video versions are available.	
Spell'n Fix by Peter Stark	6929
Write'n Spell by Peter Stark	7511
All-In-One, Spell'n Fix, and Write'n Spell package	195.00
Dynamite Disassembler	60.00
SUPER SLEUTH Disassembler System ($101.00 for OS-9 version)	99.00

DISK DRIVES 30 day guarantee, SD/DD capability	1 head	flippy	2 heads
5-1/4", 40 tracks	250.00	290.00	325.00
5-1/4", 80 tracks	325.00	335.00	425.00
MPI — Service Manual (General, covers 40 and 80 track)			20.00
MPI — Service Manual (Specify 40 or 80 track)			25.00
Siemens Manual			10.00
8", 77 tracks	375.00		495.00
8", 77 tracks, Thin-Line	450.00		525.00
Microtime II Calendar and Clock Board			75.00
Data Mart 16K EPROM bareboard (2708 chips)			30.00

SMOOTH™ and **ELEKTRA™** are trademarks of AAA Chicago Computer Center
FLEX and **UniFLEX** are trademarks of Technical Systems Consultants, Inc.
OS-9 and **BASIC09** are trademarks of Microware Systems Corp.
GIMEX® and **GHOST®** are registered trademarks of GIMIX Inc.

ELEKTRA™ SS50 Computer Products

ELEKTRA D-5 Dual drive cabinet for 5-1/4" drives with power supply, line cord, fuse, power switch, and power cable to drives	125.00
ELEKTRA HD-5 (Heavy duty version of D-5 package above)	150.00
ELEKTRA SHD-5 (Super heavy duty: Powers 1 Winchester and 1 floppy)	175.00
5" ribbon cable for dual 5 1/4" disk drives	40.00
ELEKTRA D-8 Dual drive cabinet, power supply, ps cable for 8" drives	350.00
Cabinet for dual 8" drives only	250.00
Power supply for dual 8" drives only	120.00
PS cables only (Specify brand and type of 8" drives)	30.00
8" ribbon cable for dual 8" disk drives	45.00

TERMINALS

Hazeltine 1420	495.00
Hazeltine Espirit 1	525.00
Adds Viewpoint (Green Screen)	525.00
Televideo 925 (Green Screen)	775.00
Televideo 950 (Green Screen)	995.00

Printers

Okidata ML 82A (120 cps, 9x9, bidirectional, serial and parallel)	475.00
Tractor for ML 82A	80.00
Okidata ML 83A (120 cps with tractor)	775.00
2K byte buffer (High speed RS-232)	125.00
Dot addressable graphics	50.00
Okidata ML 84 (200 cps, 2K, Graphics, Parallel)	1195.00
Okidata ML 84 (200 cps, 2K, Graphics, Serial)	1295.00
TI 810 w/lower case and full vertical forms control (limited quantity)	1400.00
Florida Data (600 cps)	3495.00
NEC 3510 Std	1495.00
NEC 7710 Std	2445.00
Epson MX-80 T (Centronics compatible parallel interface)	525.00
(with Serial RS-232 interface option)	add 75.00
Spare Print Head	39.95
Spare ribbon cartridge	8.50
Epson MX-100	725.00
C. Itoh Prowriter 8510 (Parallel Interface, 120 cps)	525.00

Optimal Technology, Inc. EP-2A-79 Eprom Programmer	169.00
(Personality Modules extra for above programmer)	
Optimal Technology, Inc. 30 pin parallel I/O board for EP-2A-79	35.00
Software package for EP-2A-79 (Specify 6800 or 6809)	30.00

Smoke Signal Broadcasting	
DCB-4A Double density Controller Board for 5" and 8" with DOS	549.00
DOS69D DOS Update with Editor and Assembler (Specify 5" or 8")	75.00
SSB Monitor (Specify 6800/6809, $8008/$E008/$F7E8)	75.00
SSB version of FLEX™ (without Editor and Assembler)	150.00
LMB-1A Motherboard	399.00
SCB-69 6809 CPU Board	399.00
PAR-1 Dual Port Parallel Board	89.00
SER-2 Dual Port Serial Board with 2 Cables	129.00
Static Memory Boards M-16-X 195.00 M-24-X 295.00 M-32-X 395.00	
Dynamic Memory Boards M-128-X 995.00 M-256-X 1295.00 M-512-X 1895.00	

GIMIX

6800 CPU Board	224.03
with timers	288.06
with baud rate option	add 30.00
with 2MHz option	add 15.00
2 MHz 6809 Plus CPU, time of day clock, battery backup, 1K NMOS RAM	578.05
CMOS RAM substitution	8.00
GIMIX Dynamic Address Translator	35.00
SWPTC compatible DAT	15.00
9511A Arithmetic Processor (4MHz)	312.00
9512 Arithmetic Processor (3MHz)	265.00
GMXBUG-09 (Terminal Based) 1K scratchpad required	98.65
Manual and Source Listing only	38.62
Bootstrap Prom	30.00
Video Prom (includes bootstrap)	30.00
Filler plate for 5-1/4" drive opening	14.92
Baud rate generator board	88.93
Missing cycle detect card	38.23
Prototyping board (50 pin slot)	58.66
Prototyping board (30 pin slot)	38.33
Disk Controllers (All have data separators and can be used with either single or double headed drives)	
5" single density controller without 1771 chip	158.38
5" single density controller complete	198.48
5" and 8" single density controller complete	226.58
5" double density controller with variable precomp	298.28
DMA 5" AND 8" double density controller with variable precomp	588.68
GIMIX version of FLEX™ (without Editor and Assembler)	90.00
Double disk regulator card	68.22
Ribbon cable for two 5 1/4" disk drives (inboard)	34.96
Ribbon cable for two 8" disk drives (outboard)	44.26
8" disk ribbon cable and back panel connector set	29.25
8" disk drive cabinet with power supply	848.18
19MB Winchester and controller update	4288.90
38MB Winchester and controller update	6688.91

Memory	CMOS WITH BAT. BACKUP	NMOS NO BAT. BACKUP
64K Static RAM Board with 24K of RAM installed	N/A	348.27
64K Static RAM Board with 32K of RAM installed	518.36	398.37
64K Static RAM Board with 48K of RAM installed	N/A	518.47
64K Static RAM Board with 56K of RAM installed	728.56	578.57
64K Static RAM Board with 64K of RAM installed	798.64	638.67
16 Socket EPROM/ROM/RAM Board		238.32
8K Promboard (2708)		98.34
32K Static RAM Board with 32K of RAM installed		245.00

I/O Boards	1 port	2 port	8 port
Serial interface	88.41	128.43	318.48
Parallel interface		88.42	198.45
Cable sets for above boards (specify board)		each	22.95

Video Boards	
64 or 32 X 16	198.71
80 X 24 without RAM character generator	398.74
80 X 24 with RAM character generator	458.76
High resolution (512 X 512 dot resolution)	996.77

2MHz 6809 PLUS Computer System w/o Disk Cont.	w/#58 Cont.	w/#68 Cont.
With 56K of memory 2498.29	2988.59*	3248.49*
With 128K of memory		3796.39
With 128K and 19MB Winchester		8888.09
With 128K, 38MB Winch. dual 8"		17488.89

*Includes GMXBUG/FLEX/OS-9 software selectable

*With CMOS RAM and Battery Backup	add 150.00
Mainframe (Chassis, PS, Switches, Fan, Motherboard, Baud Rate Gen.)	1198.19

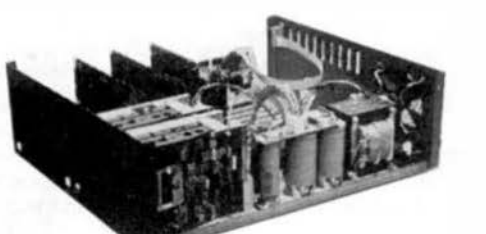
AAA Chicago Computer Center

See our ad on the previous page to your left for ordering instructions.

AAA Chicago Computer Center

ELEKTRA

COMPUTER PRODUCTS



ELEKTRA CABINET Made of heavyweight 0.090" thick aluminum. Interior is 18-1/2" wide by 21-7/8" deep by 6-3/4" high. Heavy duty A.C. line cord. A.C. fuse holder. EMI filter. Fan with filter. Back panel has 10 cutouts for 'D' type data connectors. Front panel has key on-off power switch, 2 illuminated push button switches (Reset and NMI/Abort), and two cutouts for 5-1/4" disk drives.

POWER SUPPLY Highest quality linear power supply CONSERVATIVELY rated at 15a @ 8v, 3a @ 16v, 3a @ -16v. 3 primary inputs for light, rated, and heavy loading.

DISK REGULATOR BOARD WITH CABLES Standard version for 2 floppy drives. Heavy duty version for 1 Winchester drive and 1 floppy drive.

ELEKTRA MOTHERBOARD Heavyweight 0.125" thick, 18" long by 9" wide. 11 memory (50 pin) slots. 4 or 8 slots may be cut off for shortening to 14" or 10" lengths respectively. 8 I/O (30 pin) slots. Complete address decoding of I/O slots. Choice of 4, 8, or 16 addresses per I/O slot. Base address for I/O slots can be placed at 32, 64, or 128 byte increments respectively. 1" spacing between all memory and I/O slots. Extended addressing capability for both memory and I/O ports for meeting SS-50C bus specifications. On board baud rate generator with low and high ranges providing jumper selectable rates of 75 through 38,400 for each of the five baud rate lines. Slow device circuitry permitting 1MHz 30 pin disk controllers to run with 2 MHz 50 pin CPU boards.

ELECTRA CPU 8/9 Use either the 6802 or 6808 (to run 6800 software) or 6809. Has provision for up to 3 2716 Eproms, 1K scratchpad, MC6840 triple timer, and an optional baud rate generator providing baud rates from 110 through 38,400 baud in two user selectable ranges. The board supports DMA by either HALT or BUSREQ when a 6809 CPU is used. The board does not support a DAT and therefore does not support extended addressing. The board will run any of the MIKBUG™ compatible monitors in the 6802/6808 mode and SBUG-E, HUMBUG, and GMXBUG-09 in the 6809 mode. The ELEKTRA CPU 8/9 will run any of the popular disk controller boards with the appropriate software. OS-9™ Level 1 is available as shown below.

ELEKTRA DPS DUAL PORT SERIAL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 addresses per port with the B port 2 addresses higher than the A port or for 16 addresses per port with the B port 4 addresses higher than the A port. Each port is terminated at two 16 pin dip sockets, one socket configured for modem and the other for terminal or printer. RTS, CTS, DTR, DCD, and DTR are appropriately implemented. Each port has independent selection of baud rate. Each port allows the interrupt request to be independently jumpered to the IRQ or FIRQ/NMI bus line.

ELEKTRA DPP DUAL PORT PARALLEL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be used in either the 4 or 16 addresses per I/O slot configuration occupying the first four addresses of the I/O slot. The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals. The interrupt request lines for each port may be individually jumpered to either the IRQ or FIRQ/NMI bus line.

ELEKTRA CHASSIS Includes cabinet, 110v power supply, power supply cables, standard disk regulator board with power cables, assembled motherboard with gold square pin connectors	$825.00*
ELEKTRA CHASSIS with 2 GIMIX 32K memory boards (64K total)	1300.00*
ELEKTRA CHASSIS, 2 32K GIMIX memory boards, ELEKTRA 8/9 CPU	1550.00*

The $1550.00 combination above with disk controller board and software as indicated (monitor must be added):

	SSB DOS	FLEX™	OS-9™L1
SSB DCB-4A (SD/DD, controls up to 8 floppies)	2095.00	2245.00***	2295.00***
***Includes SSB DOS also			
GIMIX #68 DMA Controller	N/A	2225.00*	N/A
GIMIX #28 Double Density, PIO Controller	N/A	1935.00*	N/A
GIMIX #58 Single Density, PIO Controller	N/A	1865.00*	1975.00*

Other combinations available

ELEKTRA CABINET	260.00
ELEKTRA 110v POWER SUPPLY	175.00*
ELEKTRA CABINET, 110v POWER SUPPLY AND POWER SUPPLY CABLES	410.00*
ELEKTRA CABINET, 110v PS, PS CABLES, STD. DISK REGULATOR AND CABLES	460.00*
*Add $30.00 for 220v	
STANDARD DISK REGULATOR BOARD AND CABLES	50.00
HEAVY DUTY DISK REGULATOR BOARD AND CABLES	75.00
FILLER PLATE FOR 5-1/4" DRIVE OPENING	10.00
FAN FILTER	10.00

See our ad on the next page for our ELEKTRA floppy drive cabinets

GOLD 10 PIN CONNECTORS (Specify male w/square pins or female)	1.50
TIN 10 PIN CONNECTORS (Specify male w/square pins or female)	.50

	BARE BOARD	KIT	ASSEMBLED
ELEKTRA MOTHERBOARD (Gold connectors)	80.00	320.00	380.00
ELEKTRA MOTHERBOARD (Tin connectors)		240.00	300.00
ELEKTRA CPU 8/9	50.00	225.00	275.00
ELEKTRA CPU 8/9 with baud rate option		250.00	300.00
ELEKTRA DPP DUAL PORT PARALLEL BOARD**	20.00	60.00	80.00
ELEKTRA DPS DUAL PORT SERIAL BOARD**	20.00	60.00	80.00
**CABLE FOR DPP OR DPS (2 needed, specify board)			20.00

MODEMS	MANUAL ANSWER	AUTO ANSWER
1200 Baud (120 cps), direct connect (U.S. Robotics)	449.00	549.00
300 Baud (30 cps), direct connect (U.S. Robotics)	179.00	219.00
300 Baud (30 cps), acoustic (U.S. Robotics)	149.00	N/A
1200 Baud (120 cps), direct connect (Hayes Smart Modem)		239.00*
300 Baud (30 cps), direct connect (Hayes Smart Modem)		589.00*

WARNING AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled kits. AAA Chicago Computer Center does warranty and maintain service for our assembled boards. The customer should carefully take into consideration the small differential separating our kit and assembled prices when making his choice of purchase.

We have introduced our line of computer equipment with the purpose of offering the highest quality of components possible at affordable prices. These products are intended for OEM applications where it is the responsibility of the purchaser to integrate these components with suitable memory, disk controllers, drives, and software along with I/O terminals to form working computer systems.

Prices and inventory are subject to change without advance notice
This ad is our catalog.

SUPER MODEM PROGRAM

Transmit manually to distant computer

Transmit disk files (text) of any length to distant computer

Receive and save disk files (text) of any length on local disk system. If sending computer does not support an X-on/X-off protocol, then the received files are limited in size by the computer memory.

Tested to transmit and receive text at speeds up to 9600 baud. (CRT terminal must be capable of operating at a baud rate higher than the one the modem is operated at.)

Half duplex option in case distant computer doesn't echo.

Echo option so user can simulate a time sharing system. (Super Modem Program doesn't support auto-answer but the source is provided for those individuals who wish to adapt our program to their special needs.)

Replaces CR with CR/LF (user option) for those using time sharing systems that don't transmit LF's.

Slow disk file transmit (user option) based on character verification for use on time sharing systems to which disk files cannot be sent at speed suggested by the baud rate.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8".
Manual and disk with both source and object code $75.00

STANDARD MODEM PROGRAM

Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option, nor X-on/X-off option. Reception of disk files is limited to those small enough to completely fit within the receiving buffer.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8".
Manual with instructions, source listing, and flow chart; disk with both source and object code $45.00
Manual with instructions, source listing, and flow chart 25.00

TERMS Minimum order $20.00 Shipping and handling estimates within the Continental U.S., add 3% (MINIMUM $2.50). Illinois residents add 6% sales tax. We will refund your overestimated shipping and handling charges. Foreign shipping and handling, add 10% (MINIMUM $10.00). Foreign orders must be prepaid in U.S. dollars. Heavy foreign items will be shipped air freight collect. Please phone between 4 PM and 6 PM weekdays if questions arise regarding shipping fees. Master Charge, Visa, and American Express honored.

Our apology. We are not staffed to answer technical inquiries through the mail. Please phone for technical help during the hours indicated above. The too frequent changing of our inventory and prices makes it uneconomical to publish a catalog. Our ads are intended to serve that purpose. Prices and inventory are subject to change without advance notice.

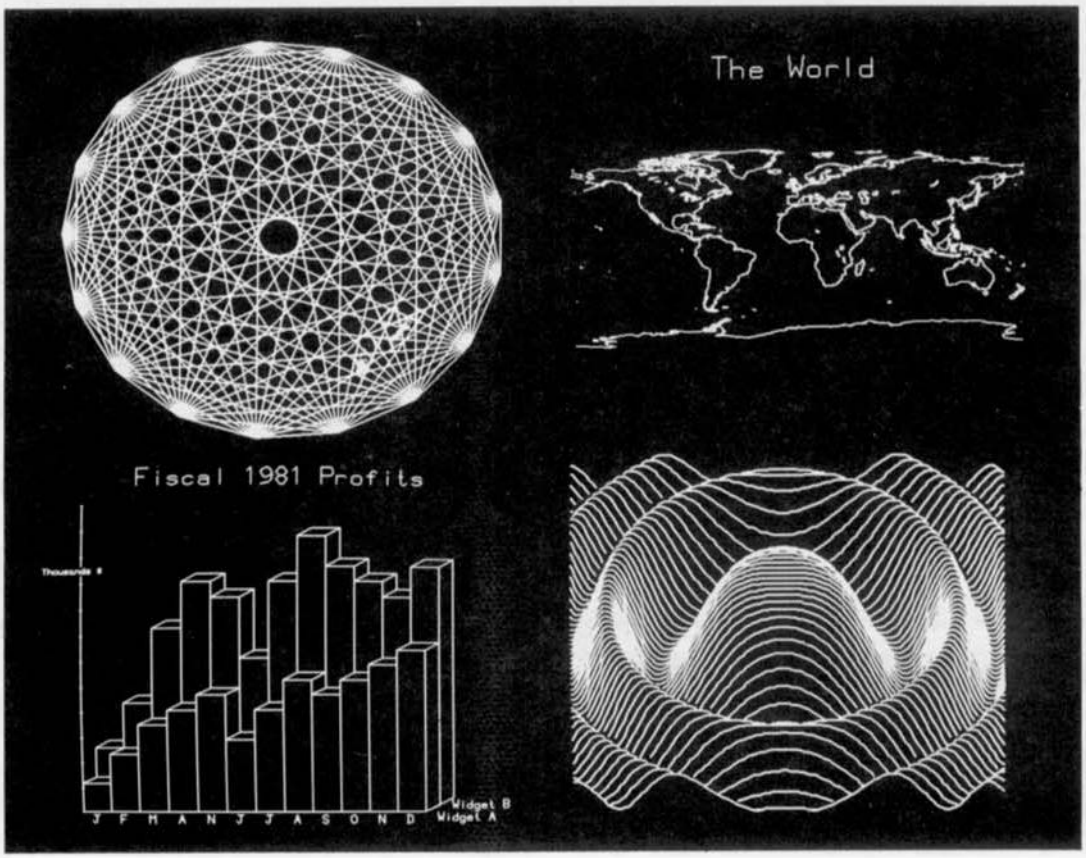
AAA CHICAGO COMPUTER CENTER
120 CHESTNUT LANE • WHEELING, IL 60090
(312) 459-0450
Technical consultation available 4 PM to 6 PM most weekdays. Closed evenings and weekends.

ANNOUNCING ElectroScreen™

only $595

the Superior Alternative

to the Traditional Alphanumeric Terminals



The ElectroScreen™ Intelligent Graphics Board Features:

Graphics

- 512 x 480 resolution bit-mapped display
- Interleaved memory access — fast, snow-free updates

Intelligence

- 6809 on-board mpu
- 6K on-board firmware
- STD syntax high level graphics command set
- Removes host graphics software burden
- Flexible text and graphics integration
- Multiple character sizes
- User programs can be run on-board

Terminal

- Terminal emulation on power-up
- 83 characters by 48 lines display
- Easy switching among user-defined character sets
- Fast hardware scrolling

Additional Features

- SS-50C and SS-64 compatible board
- Board communicates with host through parallel latches
- Composite and TTL level video output
- 8 channel 8 bit A/D converter
- Board occupies 4 address bytes

See your dealer today!

The ElectroScreen manual is available for $10. credited toward purchase of the board.

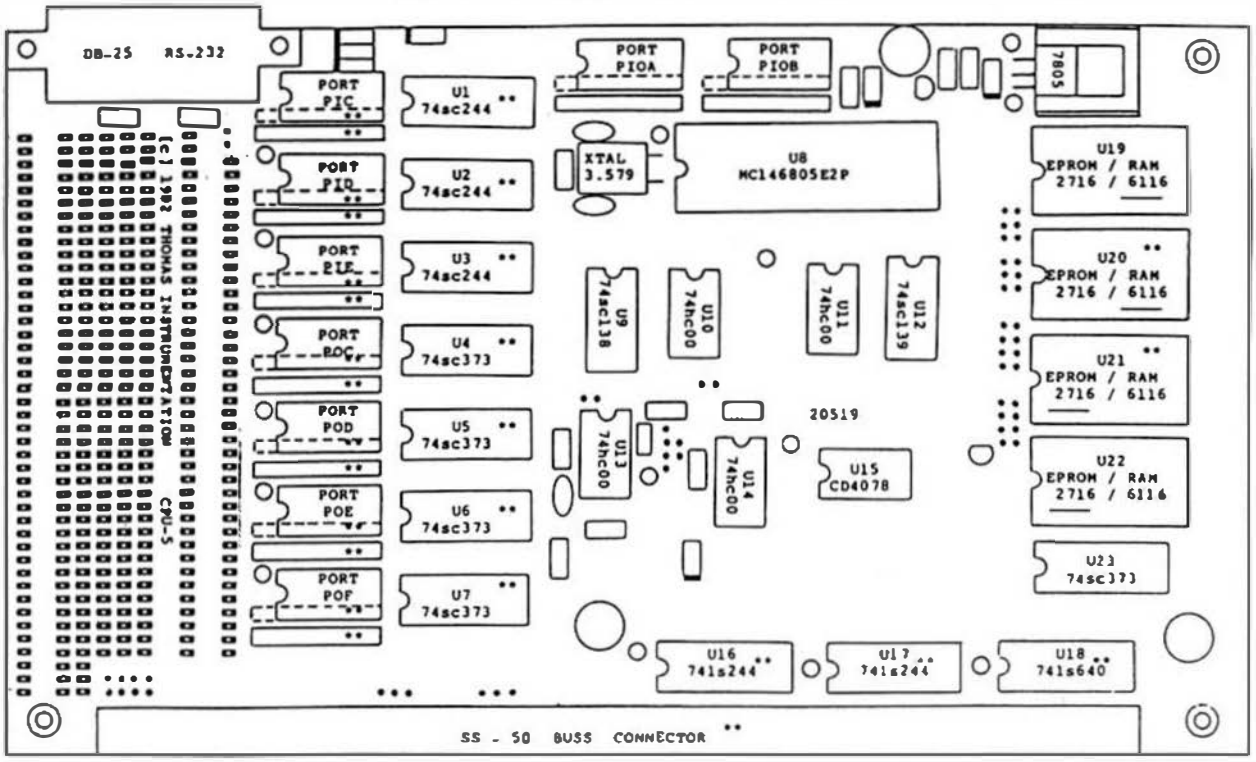
The ElectroScreen has a 90 day warranty from purchase date.

Dealers, please contact us for our special introductory package

Privac Inc (703) 671-3900

3711 S. George Mason Dr., Falls Church, Va. 22041

*** NEW 6805 ALL CMOS CARD ***



CPU-5-B	Blank card with documentation	$49.00
CPU-5-P	Partial card (parts marked ** not incl.)	$229.00 } Includes EPROM Monitor
CPU-5-F	Full card with all parts shown (fully assembled)	$349.00 } and 115 vac Power Pak.
CROSS-ASSEMBLERS	for FLEX ((c) TSC) 6809 system	$150.00

SOFTWARE

VDISK Treat extended memory as a super fast disk drive

6809 source & object	$149.00
6809 object	$ 99.00

OUTSIDE MODEM PROGRAM incl. source

UniFLEX version	$100.00
FLEX version 6800 & 6809	$ 50.00
HAYES SMART MODEM	$249.00

CROSS ASSEMBLERS for 6800,6801,6805 runs on 6809 FLEX $150.00

TV-EDIT Screen oriented Editor for FLEX 6809 $ 95.00
Please specify 5" or 8" Disk when placing order.

CHIPS

Please note when ordering IC's that a $100.00 Minimum is in effect Orders less than the minimum will be charged an additional $10.00 for handling

6116P-3	2MHZ	$13.50
2016P-2	2MHZ	$10.50
2716	1MHZ	$ 6.50
2716-1	2MHZ	$16.50
146805E2P		$25.00
68B02		$10.00
68B09		$25.00
68B21		$ 6.50
68B40		$12.00
68B50		$ 6.75
74LS640		$ 3.25

HARDWARE

NEW MEMORY PRICES TO INTRODUCE VDISK

S-R/R without memory chips	$120.00
with 8K NMOS@2MHZ	$169.00
with 16K NMOS@2MHZ	$199.00
with 32K NMOS@2MHZ	$299.00
with 48K NMOS@2MHZ	$399.00
Bare Card	$ 49.00

Extender Cards assemb. with logic aid

SS-50/50C	$ 35.00
SS-30	$ 25.00
SP-1 Prototype Board A/T	$195.00
SP-1 Bare Card	$ 49.00
SS-50 Wire Wrap Board B/C	$ 39.00
SS-30 Wire Wrap Board B/C	$ 20.00
6802 Super CPU A/T	$235.00
6802 Super CPU B/C	$ 59.00
Video RAM, B/C $49.00, A/T	$195.00
Parallel I/O, B/C $49.00, A/T	$139.00
SS-50 Backplanes 4,6,8,12,16 position @ $5.00 per slot w/o connectors	
SS-30 8 pos. BP w/o connect.	$ 39.00
Transition Card $49.00, A/T	$ 95.00
Molex Gold Male $1.60, Female	$ 1.60
Molex Tin Male $ .40, Female	$ .30

THOMAS INSTRUMENTATION

168 EIGHTH STREET — AVALON, N.J. 08202 (609) 967-4280
NJ RES. INCLUDE 5% SALES TAX
Allow 3 Weeks for Checks to Clear
CONT. USA INCLUDE $3.00 SHIPPING, CANADA $6.00, FOREIGN $12.00
MASTERCARD, VISA, and C.O.D. ACCEPTED
Send $20.00 to receive full documentation, schematics, & source listings for all boards currently in production.

'68' MICRO JOURNAL ADVERTISERS INDEX

This Index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

Statement of Ownership, management and Circulation (Act of August 12, 1970, Title 39, United States Code). 1. Title of Publication: 68 Micro Journal 2. Publication no: 468510: 3. Frequency of Issue: Monthly: A. No. of Issues Published Annually: 12, B Annual Subscription Price: $24.50 4 and 5: Complete Mailing Address of Known Office, Headquarters or General Business Offices of the Publisher: 5900 Cassandra Smith Rd., Hixson, TN. 37343 6. Full Name and Complete Mailing Address of Publisher: Donald M. Williams Sr., 5900 Cassandra Smith Rd., Hixson, TN 37343, Editor: Donald M. Williams Sr., 5900 Cassandra Smith Rd., Hixson, TN 37343., Managing Editor: Larry E. Williams, 5900 Cassandra Smith Rd., Hixson, TN 37343 7.Owner: Computer Publishing Inc., 5900 Cassandra Smith Rd., Hixson, TN 37343, whose Stockholders are: Donald M. Sr., Frances J., Larry E., Mary E., Thomas E. Williams, B. Known Bondholders, Mortgagees, and other Security Holders Owning or Holding 1 Percent or more of Total Amount of Bonds, Mortgages of other Securities: None 9.For Completion By Nonprofit Organizations Authorized To Mail At Special Rates: N/A 10. Extend and Nature of Circulation: A. Total No. Copies (Net Press Run) Average No. Copies Each Issue During Preceding 12 Months: 8660., Actual No. Copies Of Single Issue Published Nearest To Filing Date: 9230, B. Paid Circulation: 1. Sales Through Dealers And Carriers, Street Vendors And Counter Sales: Average No. Copies Each Issue During Preceding 12 Months: 3506. Actual No. Copies Of Single Issue Published Nearest To Filing Date: 3750. 2. Mail Subscriptions: Average No. Copies Each Issue During Preceding 12 Months: 4751. Actual No. Copies Of Single Issue Published Nearest To Filing Date: 4904. C. Total Paid Circulation: Average No. Copies Each Issue During Preceding 12 Months: 8257. Actual No. Copies Of Single Issue Published Nearest To Filing Date: 8654. D. Free Distribution By Mail, Carrier Or Other Means Including Samples, Complimentary, And Other Free Copies: Average No. Copies Each Issue During Preceding 12 Months: 110. Actual No. Copies Of Single Issue Published Nearest To Filing Date: 152. E. Total Distribution (Sum Of C And D): Average No. Copies Each Issue During Preceding 12 Months: 8367. Actual No. Copies Of Single Issue Published Nearest To Filing Date: 8806. F. Copies Not Distributed: 1. Office Use, Left Over, Unaccounted,Spoiled After Printing: Average No. Copies Each Issue During Preceding 12 Months: 293, Actual No. Copies Of Single Issue Published Nearest To Filing Date: 424. 2. Return From News Agents: Average No. Copies Each Issue During Preceding 12 Months: 0, Actual No. Copies Of Single Issue Published Nearest To Filing Date: 0, G. Total (Sum of E, F1 and 2-Should Equal Net Press Run Shown in A: Average No. Copies Each Issue During Preceding 12 Months: 8660, Actual No. Copies Of Single Issue Published Nearest To Filing Date: 9230, 11. I Cerrtify That The Statements Made By Me Above Are Correct And Complete. (Signed): Larry E. Williams, Managing Editor,68 Micro Journal. Computer Publishing, Inc. 12. For completion by publishers mailing at the regular rated. Permission requested.

THE CHIEFTAIN™ 5¼-INCH WINCHESTER HARD DISK COMPUTER

SO ADVANCED IN SO MANY WAYS . . . AND SO COST-EFFECTIVE . . . IT OBSOLETES MOST OTHER SYSTEMS AVAILABLE TODAY AT ANY PRICE.

● HARD DISK SYSTEM CAPACITY

The Chieftain series includes 5¼- and 8-inch Winchesters that range from 4- to 60-megabyte capacity, and higher as technology advances. All hard disk Chieftains include 64-k memory with two serial ports and DOS69D disk operating system.

● LIGHTNING ACCESS TIME

Average access time for 5¼-inch Winchesters is 70-msec, comparable to far more costly hard disk systems. That means data transfer ***ten-times faster*** than floppy disk systems.

● 2-MHZ OPERATION

All Chieftains operate at 2-MHz, regardless of disk storage type or operating system used. Compare this to other hard disk systems, no matter ***how*** much they cost!

● DMA DATA TRANSFER

DMA data transfer to-and-from tape and disk is provided for optimum speed. A special design technique eliminates the necessity of halting the processor to wait for data which normally transfers at a slower speed, determined by the rotational velocity of the disk.

● RUNS UNDER DOS OR OS-9

No matter which Chieftain you select . . . 5¼- or 8-inch floppy, or 5¼- or 8-inch Winchester with tape or floppy back-up . . . they ***all*** run under DOS or OS-9 with ***no need*** to modify hardware or software.

● UNBOUNDED FLEXIBILITY

You'll probably never use it, but any Chieftain hard disk system can drive up to 20 other Winchesters, and four tape drives, with a single DMA interface board!

● SMOKE SIGNAL'S HERITAGE OF EXCELLENCE

This new-generation computer is accompanied by the same ***Endurance-Certified*** quality Dealers and end-users all over the world have come to expect from Smoke Signal. And support, software selection and extremely competitive pricing are very much a part of that enviable reputation.

20-Megabyte Tape Streamer Back-Up Option

Available with all Chieftain hard disk configurations. This cartridge tape capability provides full 20-megabyte disk back-up in less than five minutes with just one command, or copy command for individual file transfers. Transfers data tape-to-disk or disk-to-tape. Floppy back-up is also available in a variety of configurations.

The Chieftain Computer Systems:

Here are the Chieftain 6809-based hard disk computers that are destined to change the data processing industry . . .

☐ **CHIEFTAIN 95W4**
4-megabyte, 5¼-inch Winchester with a 360-k floppy disk drive (pictured).

☐ **CHIEFTAIN 95XW4**
4-megabyte, 5¼-inch Winchester with a 750-k octo-density floppy disk drive.

☐ **CHIEFTAIN 98W15**
15-megabyte, 5¼-inch Winchester with a 1-megabyte 8-inch floppy disk drive.

☐ **CHIEFTAIN 9W15T20**
15-megabyte, 5¼-inch Winchester with a 20-megabyte tape streamer.

Write or call today for details (including the remarkably low prices) on the total Chieftain Series . . . and on dealership opportunities.

"See us at Booth 323—Comdex Fall"

SMOKE SIGNAL BROADCASTING®
31336 VIA COLINAS
WESTLAKE VILLAGE, CA 91362
TEL (213) 889-9340

Name _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone ( ) _____

2MHZ 6809 SYSTEMS

GIMIX offers you a variety to choose from!

38 MB WINCHESTER SYSTEM $17,498.99

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 512KB Static RAM
- ★ 8 RS232C Serial Ports
- ★ 2 Parallel Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ Dual 8" DSDD Floppy Disk System
- ★ Dual Winchester Subsystem with Two19 MB 5¼" Winchester Drives

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Debugger
- ★ OS-9 Text Editor
- ★ OS-9 Assembler

19 MB WINCHESTER SYSTEM $8998.09

HARDWARE FEATURES:

- ★ 128K Static Ram
- ★ 2MHz 6809 CPU
- ★ 19 MB 5¼" Winchester DMA Subsystem
- ★ 4 RS232C Serial Ports
- ★ 1 MB 5¼" Floppy Disk Drive
- ★ DMA Double Density Floppy Disk Controller

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Text Editor
- ★ OS-9 Debugger
- ★ OS-9 Assembler

128KB MULTI-USER SYSTEM $6997.39

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ DMA Double Density Floppy Disk Controller
- ★ 128KB Static Ram
- ★ 2 RS232C Serial Ports
- ★ Dual 8" DSDD Floppy Disk System

SOFTWARE FEATURES: Your choice of either UniFLEX or OS-9 LEVEL TWO. Both are Unix-like Multi-User/Multi-Tasking Operating Systems.

56KB FLEX / OS-9 "SWITCHING" SYSTEM $4148.49

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 56K Static Ram
- ★ 2 RS232C Serial Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ 2 Built-in 5¼" 40tr DSDD Disk Drives (80 Track DSDD Drive Option . . add $400.00)

SOFTWARE FEATURES:

- ★ GMXBUG monitor — FLEX Disk Operating System
- ★ OS-9 LEVEL ONE Multi-tasking operating system for up to 56K of memory

WINCHESTER SUBSYSTEMS

Winchester packages are available for upgrading current **GIMIX** 6809 systems equipped with DMA controllers, at least one floppy disk drive, and running FLEX, OS-9 LEVEL ONE or OS-9 LEVEL TWO. The packages include one or two 19MB (unformatted) Winchester drives, DMA Hard Disk Interface, and the appropriate software drivers. The Interface can handle two 5¼" Winchester Drives, providing Automatic Data Error Detection and Correction: up to 22 bit burst error detection and 11 bit burst error correction. **UniFLEX NOW AVAILABLE**

Dual drives can be used together to provide over 30 MBytes of on line storage -- or use one for back-up of the other. (More convenient and reliable than tape backup systems.

#90 includes one 19MB Drive, Interface, and Software **$4288.90**
#91 includes two 19MB Drives, Interface, and Software **$6688.91**

Contact GIMIX for systems customized to your needs or for more information.

50 HZ Export Versions Available

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

FLEX & RS COLOR COMPUTER

If you are tired of playing games on your TRS-80C™ Color Computer, or find that you are handicapped by the limitations of the RS BASIC in trying to write a Program that will allow you to actually USE the Color Computer as a COMPUTER; and if you have been studying the Advertisements in this Magazine and wishing that you could run THESE Programs on your Computer, YOU ARE READY TO MOVE UP TO THE FLEX9™ Operating System. If you want to have REAL PROGRAMMING POWER, using an Extremely Powerful Business BASIC, PASCALs, C Compilers, a full-blown Macro Assembler with a Library capability so you are not continuously "reinventing the wheel", YOU ARE READY TO MOVE UP TO THE FLEX9™ Operating System. If you would like to see if YOU REALLY COULD USE A COMPUTER IN YOUR BUSINESS, or begin to make your Computer start PAYING IT'S OWN WAY by doing some Computer Work for the millions of small businesses around you, such as Wordprocessing, Payroll, Accounting, Inventory, etc., then YOU ARE READY TO MOVE UP TO THE FLEX9™ Operating System. How?? DATA-COMP has the way!

DATA-COMP's FLEX9™ Conversion for the TRS-80C™ Color Computer was designed for the SERIOUS COMPUTER USER: with features like greatly increased Display Screens, WITH Lower Case Letters, so you can put a FULL Menu on ONE Screen, or see SEVERAL Paragraphs at the same time; with features like providing a FULL Keyboard so you have FULL Control of your Computer AND it's Programs NATURALLY, without needing a chart to see what Key Combination will give you what function; with USER ORIENTED functions to make using the Operating System natural, like having the Computer AUTOMATICALLY determine what type of Disk is being used in what type of Disk Drive and working accordingly, rather that you have to specify each and every thing for it, or like having the Computer work with the Printer you have been using all along without you having to tell the new Operating System what is there; etc., etc., etc.

DATA-COMP has everything you need to make your TRS-80C™ Color Computer WORK for YOU; from Parts and Pieces to Full, Ready To Use SYSTEMS. DATA-COMP designs, sells, services, and SUPPORTS Computer SYSTEMS, not just Software. CALL DATA-COMP TODAY to make your Computer WORK FOR YOU!

-- SYSTEM REQUIREMENTS --

FLEX9 Special General Version w/Editor & Assembler (which normally sell for $50.00 ea.)	$150.00
F-MATE(RS) FLEX9 Conversion Rout. for the RS Disk Controller when purchased with Special General FLEX9 Sys.	$69.95
when purchased without the General FLEX9 Sys.	$79.95
NEW -- Full Source Code for the Conv. Routines	$159.95
Set of Eight 64K RAM Chips w/ Mod. Instructions	$99.95
Color Computer with 64K RAM and EXT. BASIC	$549.95
Color Computer with 16K RAM	$375.95
Color Computer with 16K RAM and EXT. BASIC	$465.95

-- SPECIAL SYSTEM PACKAGES --

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, Special General Version of FLEX9™, F-MATE(RS)™, and a Box of 10 Double Density Diskettes; a COMPLETE, ready to run SYSTEM on your Color TV Set.
$1749.95

Now Available
Enhanced F-MATE Version 2.0

-- DISK DRIVE PACKAGES, with RS Controller --
These Packages include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable:

PAK #1 ==> 1 Single Sided, Double Density Sys.	$499.95
PAK #2 ==> 2 Single Sided, Double Density Sys.	$729.95
PAK #3 ==> 1 Double Sided, Double Density Sys.	$579.95
PAK #4 ==> 2 Double Sided, Double Density Sys.	$889.95

-- PARTS AND PIECES --

Radio Shack Disk Controller	$169.95
1 ea. Single Sided, Double Density Disk Drive	$249.95
1 ea. Double Sided, Double Density Disk Drive	$349.95
Single Drive Cabinet with Power Supply	$79.95
Double Drive Cabinet with Power Supply	$99.95
Single Drive Disk Cable for RS Controller	$24.95
Double Drive Disk Cable for RS Controller	$34.95
Micro Tech. Prods., Inc. LOWER CASE ROM Adapter	$74.95
Radio Shack BASIC Version 1.1 ROM	$34.95

SOFTWARE

CHESS 6809

Requires FLEX™ and one of the following CRT terminals

Now Runs On Any Typ Terminal

Features:

- Two display boards.
- Four levels of play.
- Point scoring system.
- Play white or black.
- Change or set-up board/piece positions.
- Forfeit move.
- Swap sides.
- Make move and swap sides.
- Change skill level.
- Stop and restart game.
- Solve 'Mate in 1-2-3-4' moves.

$79.95 Specify 5" or 8" disk

This is one of the strongest CHESS programs running on any microcomputer, estimated USCF Rating 1500 +.

Note: Personal checks allow 3-4 week delivery.

DIET-TRAC Forecaster

A Diet Planning and Analysis Program

DIET-TRAC Forecaster is a program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C.P.F %) or grams of Carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individuals are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's predicted weight for a 30-day period is printed.

FLEX VERSION	**$59.95**
UniFLEX VERSION	**$89.95**

PRINTERS

The Epson MX-80
$495.00

The Epson MX-100
$725.00

MX-70 $355.00 MX-80 FT $575.00

MEMORY

SWTPC-Motorola, MP32
32K Dynamic Memory Board
Assembled & Tested
1 MHZ - No extended addressing
Can be set up for $0-7FFF or 8000-FFFF

$149.95

MasterCard VISA

DATA-COMP
SOUTH EAST MEDIA

P.O. Box 794 *HIXSON, TN 37343*

1-615-842-4601

Verbatim Diskettes.

5" Soft Sector Disks

Single Side Single Density	$2.75 ea.
Single Side Double Density	$2.75 ea.
Double Side Double Density	$4.92 ea.
Plastic Storage Box	$2.00 ea.

8" Soft Sector Disks

Single Side Single Density	$3.75 ea.
Single Side Double Density	$4.10 ea.
Double Side Double Density	$4.75 ea.
Plastic Library Box	$5.00 ea.

Foreign Orders Add 10% Surface—20% Air Mail

DRIVES & CABINETS W/PS

5 1/4"

TANDON - Single Sided, Double Density, 40 Track.....$249.95

TANDON - Double Sided, Double Density, 40 Track.....$349.95

CABINET - Single Drive with Power Supply.....$ 79.95

CABINET - Double Drive with Power Supply.....$ 99.95

CABLE - Single Drive.....$ 24.95

CABLE - Double Drive.....$ 34.95

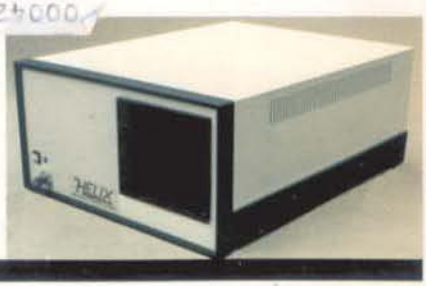
* NOTE - When ordering cables please specify S50 Bus or Other!!!

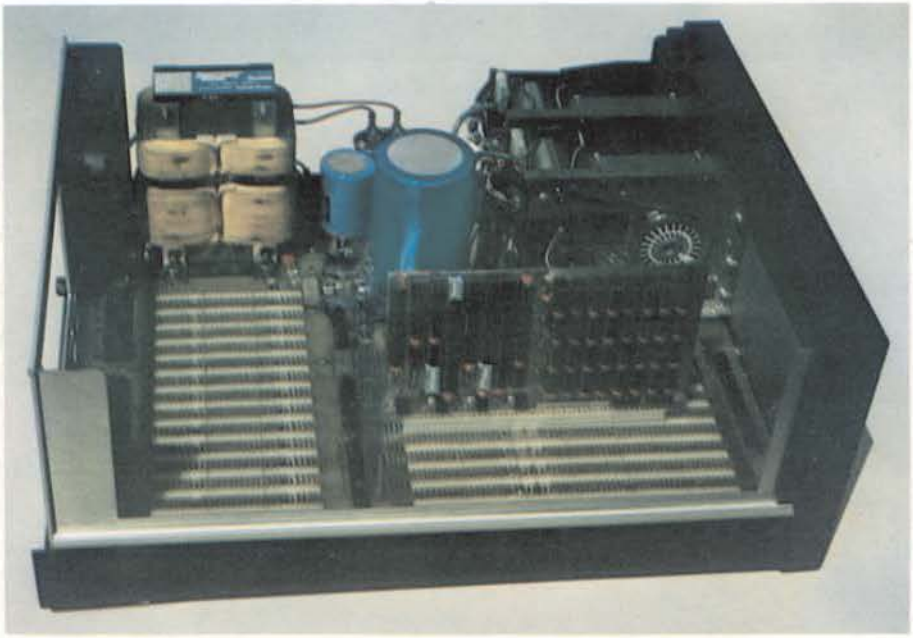
Call or write for disk controller Board information.

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

HELIX™





THE MAINFRAME

- Industry Standard Optima™ Cabinet
- Largest Constant Voltage Power Supply in the industry
- S-64 Bus gives 16 Bit Power and S-50 Bus Compatibility
- 10 Main (S-64) Slots
- 14 I/O (S-30) Slots plus 2 On-board
- On-board Baud Rate Generator to 38.4Kb
- Space and Power for two 5¼" Disk Drives
- Full Address Decoding for I/O Slots
- Two RS-232 Serial and Two parallel Ports On-board
- Single Board Construction for Reliability
- Faraday Shielded Bus Lines give "Text Book Clean" Signals

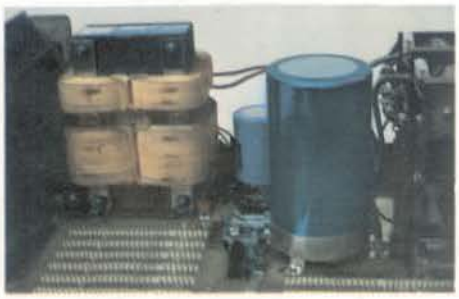
THE PROCESSORS

6809

- Standard 2 MHz Operation
- Standard DAT Compatible with GIMIX and SWTPC
- Standard 6840 Interval Timer
- Standard 1K Scratchpad RAM
- Standard Clock/Calendar with Battery
- Provision for Programmers Console

68000

- Standard 8 MHz Operation
- Memory Management Hardware
- Provision for Programmers Console
- 16 Bit Power and 8 Bit Compatibility



The HELIX™ computer system represents the latest advance in S-50 bus computer systems. Relying on the physical nature of S-50 bus connectors to guarantee compatibility, the HELIX adds 14 bus lines (becoming S-64) to allow a 68000 processor to operate with full 16 bit data transfer and 24 bit addressing, while at the same time providing full interchangability with existing S-50 components.

Offered with a selection of processors, memories, and peripheral controllers, a HELIX system can be configured for applications ranging from advanced hobbyist to multiterminal time-sharing.

Designed to offer the utmost in speed, reliability, and utility at a reasonable price, it represents a new standard of quality for those who require a professionally designed computer for professional use.

THE POWER SUPPLY

- Ferro-resonant Transformer for Line Noise and Under-Voltage Protection
- Conservative 25 Amps at 8.5 Volts
- Conservative 5 Amps at ±16 Volts
- Conservative Component Rating for Reliability

THE COMPONENTS

- Fully Socketed
- Gold Plated Bus Connectors
- Only "B" Series 68XX Components Used
- Only Top Grade Logic Circuits Used
- Industrial Grade Components Throughout

THE MEMORIES

DM-64

- Field Proven
- Proprietary Memory Control Logic
- Fully Transparent Refresh
- Tested at 2.5 MHz Operation

DM-512

- 512K Bytes on a Single S-64 Board
- 16 Bit Power and 8 Bit Compatibility
- Runs in Existing S-50 Systems where Physical Space Allows
- Full 24 Bit Addressing
- Fully Transparent Refresh

THE PRICES

Because of the variety of configurations possible, full pricing cannot be given. Representative prices are:

- 64K 6809 HELIX $1995
- 64K 68000 HELIX $2595
- 512K 6809 HELIX $4450
- 512K 68000 HELIX $4995

HAZELWOOD COMPUTER SYSTEMS

907 E. Terra, O'Fallon, Missouri 63366 (314) 281-1055

Dealer and OEM Inquiries Invited. We support our Dealers.

Optima is a Trademark of Scientific-Atlanta